

Telescience Testbed Pilot Program
Final Report
Volume III
Experiment Summaries

IN-14
DATE OVERRIDE
43019

P.142

Barry M. Leiner

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report TR-89.9
February 1989

(NASA-CR-188834) TELESCIENCE TESTBED PILOT
PROGRAM, VOLUME 3: EXPERIMENT SUMMARIES
Final Report (Research Inst. for Advanced
Computer Science) 142 p

N92-10032

CSCL 14B

Unclass

G3/14 0043019

RIACS

Research Institute for Advanced Computer Science
An Institute of the Universities Space Research Association

Telescience Testbed Pilot Program

Final Report

Volume III

Experiment Summaries

Barry M. Leiner

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report TR-89.9
February 1989

The Universities Space Research Association (USRA), under sponsorship from the NASA Office of Space Science and Applications, conducted a Telescience Testbed Pilot Program. Fifteen universities, under subcontract to USRA, conducted various scientific experiments using advanced computer and communications technologies. The goals of this pilot program were to develop technical and programmatic recommendations for the use of rapid-prototyping testbeds as a means for addressing critical issues in the design of the information system of the Space Station Freedom era.

This is the final report for the Pilot Program. It consists of three volumes. Volume I provides an Executive Summary. Volume II contains the integrated results of the program. Volume III provides summaries of each of the testbed activities.

*This work was supported in part by
Contract NASW-4234 from the National Aeronautics and Space Administration (NASA)
to the Universities Space Research Association.*

Acknowledgement

The work described herein is the result of the close cooperation of a large number of people throughout the country. Fifteen universities representing a cross-section of space science disciplines along with several NASA centers were involved in the program and contributed considerable time and effort. This report, in particular, represents the effort of approximately 50 people. These people are listed in Appendix A.

The author (really more of an editor) would like to acknowledge and express appreciation for the dedication and hard work exhibited by all involved. It is clear that the successes of the program are due to their effort.

On behalf of USRA and all of the program participants, I would like to acknowledge the support given to the program by NASA, in particular Erwin Schmerling and James Weiss of NASA Headquarters and Daryl Rasmussen of Ames Research Center. Without their efforts, this program would not have existed and been the success that we believe it to be.

I would also like to express appreciation to Maria Gallagher and Lorraine Fisher of RIACS for their long hours and hard work throughout the program and in helping to pull together this report.

THIS PAGE IS INTENTIONALLY LEFT BLANK

TTPP Final Report

Volume I: Executive Summary

Table of Contents

Acknowledgement	i
Section 1 Introduction.....	I-1
Section 2 Program Overview	I-3
2.1 Astronomy and Astrophysics.....	I-4
2.2 Earth System Sciences	I-5
2.3 Life Sciences.....	I-6
2.4 Microgravity Sciences	I-6
2.5 Telescience Technologies	I-7
Section 3 Highlights of Results	I-9
3.1 General Technical Results	I-9
3.2 Astronomy	I-11
3.3 Earth Sciences.....	I-12
3.4 Life Sciences.....	I-13
3.5 Microgravity Sciences	I-15
3.6 Programmatics	I-15
Section 4 Conclusions.....	I-17
Appendix A Program Participants	I-19
Appendix B Glossary	I-25
Appendix C Bibliography	I-29
Index	I-39

PRECEDING PAGE BLANK NOT FILMED

THIS PAGE IS INTENTIONALLY LEFT BLANK

TTPP Final Report

Volume II: Program Results

Table of Contents

Acknowledgement	i	
Section 1	Introduction and Summary	II-1
1.1	Introduction and Document Overview.....	II-1
1.2	Highlights of Results	II-2
1.2.1.	General Technical Results	II-2
1.2.2.	Astronomy	II-4
1.2.3.	Earth Sciences	II-5
1.2.4.	Life Sciences	II-6
1.2.5.	Microgravity Sciences	II-8
1.2.6.	Programmatics	II-8
Section 2	Program Description	II-11
2.1	Summary of Testbed Activities	II-12
2.1.1.	Astronomy and Astrophysics	II-12
2.1.2.	Earth System Sciences	II-13
2.1.3.	Life Sciences	II-14
2.1.4.	Microgravity Sciences	II-15
2.1.5.	Telescience Technologies	II-15
2.2	Programmatic Activities	II-16
Section 3	Discipline Summaries	II-21
3.1	Astronomy and Astrophysics Summary	II-21
3.1.1.	Teleoperation of Instruments and Experiments	II-22
3.1.2.	Teleanalysis.....	II-25
3.1.3.	Use of Networks to Enhance Programmatics	II-26
3.1.4.	Standards and Commonality	II-29
3.1.5.	Network Requirements	II-30
3.2	Earth Sciences Summary	II-32
3.2.1.	Teleanalysis.....	II-32
3.2.2.	Teledesign.....	II-33
3.2.3.	Network Infrastructure	II-34
3.2.4.	Operations	II-35
3.3	Life Sciences Summary	II-35
3.3.1.	How Life Sciences Differs From Other Disciplines	II-35
3.3.2.	Functional Areas Supporting Telescience	II-38
3.4	Microgravity Sciences Summary	II-51
Section 4	Results and Conclusions	II-55
4.1	Design and Analysis	II-55
4.2	Integrated Results on Teleoperations	II-58
4.3	Integrated Results on Network Infrastructure.....	II-64
4.4	Programmatic Lessons, Results, and Recommendations.....	II-66

Appendix A	Program Participants	II-75
Appendix B	Bibliography	II-81
Appendix C	Statement of Work	II-91
Appendix D	Subcontracts Acquisition Plan	II-95
	Announcement of Opportunity	II-98
	AO Distribution List	II-103
	Proposed PRG Membership.....	II-104
Appendix E	Glossary	II-105
Index	II-109

TTPP Final Report

Volume III: Experiment Summaries

Table of Contents

Acknowledgement	i
Section 1	
Teledesign Experiments.....	III-1
1.1 Techniques for Sharing Software and Infrared Data Among SIRTF Instrument Teams (Univ. of Arizona).....	III-1
1.2 Portability of an Ada Realtime System (Univ. of Colorado).....	III-3
1.3 Off-line Guidance Requirements (Univ. of Michigan).....	III-5
1.4 On-line Guidance (Univ. of Michigan).....	III-7
1.5 Extended Software Control System (UC Berkeley)	III-8
1.6 Software Emulation of 8085 Microprocessors (UC Berkeley).....	III-10
Section 2	
Teleoperations Experiments.....	III-13
2.1 Technology for Teleoperation (Univ. of Arizona).....	III-13
2.2 Teleoperation of an Astrometric Telescope (Univ. of Arizona).....	III-18
2.3 Teleoperation of a Fluid Handling Laboratory (Univ. of Arizona)	III-22
2.4 Acquisition of Astronomical Images From A Remote Infrared Camera (IR- Camera) (Univ. of Arizona)I	II-27
2.5 Teleoperation of a Ground Observatory (Univ. of Colorado)	III-30
2.6 Teleoperations Using the OASIS Software Package (Univ. of Colorado).....	III-33
2.7 Operations Management System (OMS) Study (Univ. of Colorado)	III-36
2.8 Packet Telemetry, Packet Command, and Standard Format Data Units for Science Instrument Operations (Univ. of Colorado).....	III-39
2.9 MIT/KSC Space Life Sciences Telescience Testbed (MIT).....	III-43
2.10 Remote Operation of Spaceborne Microgravity Materials Science Experiments (RPI)	III-51
2.11 Remote Operation of A Science Instrument (Smithsonian Astrophysical Observatory)	III-56
2.12 Remote Commanding and Telemetry Reception (UC Berkeley)	III-59
Section 3	
Teleanalysis Experiments	III-65
3.1 Telescience Field Campaign (Univ. of Colorado)	III-65
3.2 Remote Access to the SME Database (Univ. of Colorado)	III-67
3.3 Remote Data Analysis (Cornell University)	III-68
3.4 Telescience Occultation Observation (MIT)	III-71
3.5 Interactive Access to Purdue's Field Research Database (Purdue University)	III-76
3.6 Access to the University of Wisconsin's Man-Computer Data Access System (McIDAS) (Purdue University).....	III-78
3.7 Spacelab-2 Data Flow Simulation (Stanford)	III-80
3.8 Effectiveness of Optical Disk Media for Database Storage (Purdue University)	III-82

3.9	Remote Data analysis for EUVE (UC Berkeley).....	III-84
3.10	Real-time Rocket Data (UC Berkeley)	III-86
3.11	Browse Testbed (Univ. of California, Santa Barbara).....	III-87
3.12	Telescience Field Experiment (Univ. of California, Santa Barbara)	III-89
3.13	Variable Resolution Transmission of Digital Image Data (Univ. of Rhode Island)	III-90
3.14	Delivery of Real-time Meteorological Products (Univ. of Wisconsin)	III-92
Section 4	Infrastructure Experiments.....	III-95
4.1	Communications Architecture (Univ. of Michigan).....	III-95
4.2	Portable Workstations (Univ. of Michigan).....	III-96
4.3	Use of Networks To Enhance Programmatic (Smithsonian Astrophysical Observatory)	III-98
4.4	Measurement of Network Performance (Smithsonian Astrophysical Observatory)	III-99
4.5	Telescience Workstation Environment (RIACS).....	III-101
4.6	Rapid Prototyping Testbed Program (RIACS)	III-109
4.7	Telescience Workstation Activities (Stanford)	III-111
Appendix A	Program Participants.....	III-115
Appendix B	Glossary	III-121
Appendix C	Bibliography	III-125
Index	III-135

Section 1 Teledesign Experiments

1.1 Techniques for Sharing Software and Infrared Data Among SIRTF Instrument Teams (Univ. of Arizona)

This experiment involves the evaluation of the current Internet for the use of file and image transfer between SIRTF instrument teams.

Experiment Description and Summary Conclusion

This experiment evaluates the efficiency of the existing Internet for remote file transfer and database access to another institution and was conducted in collaboration with the TTPP group at Cornell University. Both groups were using Sun workstations running UNIX and IRAF V2.6. References to each machine will be by institution name (i.e. Arizona and Cornell).

IRAF, again, was chosen for the main user interface because of it's network capabilities. IRAF uses the TCP/IP network interface for the actual communications.

The experiment also investigated using Sun's Network File System (NFS) for file sharing between SIRTF team members. NFS allows Sun systems connected by a network to share files. It is used extensively at Steward for sharing software packages. We tried to carry this "file sharing" idea further. It was hoped that NFS would allow SIRTF instrument teams to share software, however, for NFS to work efficiently and safely, a sustained minimum of 9600 baud (960 bytes/second) is recommended at all times. With actual data rates fluctuating between 1 to 10 kbytes/second, the idea of mounting files systems across great distances proved not possible and therefore not presented here..

Issue Investigated

The main issue addressed was current network response times. Are they efficient enough to conduct analysis of remote image data with reasonable ease? The secondary issue addressed was that of software commonality.

Experiment Hypothesis

It was hypothesized that if the network provided adequate response times, researchers could reduce travel time to various data sites by conducting certain research tasks at their own institutions. And, by using common data storage formats, software and workstations, the number of additional programs necessary to perform remote analysis is reduced.

Method of Investigation

The coordinating efforts of both TPPP teams involved making modifications to certain UNIX files and IRAF files for IRAF communications to work. Once these modifications were made, the actual test were conducted by the Telescience member at Arizona. Use was made of the Flexible Image Transport System (FITS), a standard format in the astronomical community for the interchange of image data.

Three FITS files of varying sizes (20k, 72k, 457k), were transferred from Arizona to Cornell. At Cornell, IRAF images were made of these three files. These three images were then displayed on the Arizona machine with reasonable response times (see Experiment Results for details).

Experiment Results

Two test were conducted. The first involved simple file transfers using the UNIX utility ftp. The second mode remotely displayed an image bitmap stored on the remote machine at Cornell. All tests were conducted on 9-22-88 and 9-27-88.

1. File Transfer using ftp.

Three FITS files from Arizona were transferred to Cornell. Images names, their dimensions and types were:

- (1) 11551 [489,465] real
- (2) n253 [129,129] real
- (3) m87 [64, 64] real

Time: 7:30 am Arizona, 10:30 am Cornell

Date: 9-22-88

FITS file:	Size/Bytes	Clock Time: (seconds)	Rate: (kbytes/sec)
11551.fits	457,920	118.92	3.8
n253.fits	72,000	17.34	4.1
m87.fits	20,160	2.84	7.0

2. Image display from remote machine Cornell:

While remotely logged in and running IRAF at Comell, IRAF was instructed to display an image on the Arizona workstation.

Time: 7:05 am Arizona, 10:05 am Cornell

Date: 9-27-88

Image file:	Size/Bytes:	Clock Time: (seconds)	Rate: (kbytes/sec)
11551	953,344 .pix	90.0	10.6
n253	67,588 .pix	30.0	2.2
m87	17,408 .pix	12.0	1.4

In conclusion, transfer rates overall were better than expected, ranging from 1.4 kbytes/second to 10.6 kbytes/second. For the purpose of remote data access, network data rates of 5 to 10 kbytes as shown in this experiment are adequate only if sustainable. The observed data rates are high enough to be useful for rapid transfer of infrared data, but are currently inadequate for significant transfers of large images as would be found with optical CCD's. The overall rates over the Internet appear to have improved over the past few months. For example, an earlier attempt to conduct this experiment was aborted because of poor transfer rates of 1 to 2 kbytes/second (see Cornell TTPP Monthly Report, April 1988).

The use of a common data storage format, fits, and software, IRAF, eliminated the need for yet another program to process the image data. In this case, commonality of workstations was not as important as IRAF is available on various systems. Because of IRAF's networking capability, this same experiment would have been possible if the remote machine was running IRAF on a VMS system.

1.2 Portability of an Ada Realtime System (Univ. of Colorado)

While converting the OASIS teleoperation package to the Sun workstation under the UNIX operating system, LASP has conducted a study on the portability of an operational realtime system written in Ada and running in a hosted environment.

Experiment Description and Summary Conclusions

The Ada language was designed in particular to increase the portability of complex software systems. Whereas large non-realtime systems written in Ada have proven to be extremely portable, very little study has been done on this subject concerning full-fledged operational realtime systems. The conversion of the OASIS teleoperation package from the VAX/VMS environment to the Sun/UNIX environment provided such a case study. Preliminary results show that the portability of a realtime system can be greatly enhanced by carefully selecting the Ada compiler.

Because the delivery of the Sun workstation was very late (delivery of a partial system in August 1988 instead of February 1988 and delivery of the completed system in October 1988), this report gives only preliminary results. Final results will be published when available in a LASP technical document as a TTPP product.

Issues Investigated

To what level is a realtime system, written in Ada and running in an hosted environment (as opposed to an embedded environment), portable? 1) Can the code of a realtime system written in Ada for one system compile with little or no change on another system? 2) Is the behavior of a realtime system written in Ada reproducible with little or no change to the code from one system to another?

Experiment Hypothesis

Realtime systems such as OASIS which are written in Ada must use low-level Ada constructs (such as the ones that are defined in Chapter 13 of the Language Reference Manual-LRM) for which compiler implementors are given considerable freedom. Therefore, realtime Ada programs are not as portable (code-wise and behavior-wise) as non-realtime Ada program.

Method of Investigation

The investigation was divided into three different parts: 1) Identification of the OASIS code that was potentially non-portable; 2) Selection of Ada compiler that provided the best fit between this code and the compiler implementation; 3) Porting of the code with certification of the converted code at every step. Parts 1 and 2 have been described in two LASP internal documents and resulted in the choice of the VERDIX compiler.

Experiment Results

As mentioned above, the experiment is still in progress at the time of the writing of this document and we cannot conclude yet how the realtime behavior of OASIS is affected by the porting from the VAX to the Sun workstation (i.e. on the runtime portability).

Some of the most important lessons learned so far include:

- Portability can be facilitated by: a) carefully documenting and justifying the Ada features that need to be used in the development of a realtime system; and b) selecting the compilers that implement best those features.
- There are many issues with existing (certified) Ada compilers that prevent legitimate Ada code to either compile or execute properly. The OASIS porting has pointed out many instances where the code will compile and execute correctly with the DEC/Ada compiler but will not compile or will not execute properly with the VERDIX compiler. This problem has caused the majority of our OASIS code changes so far.
- The underlying machine architecture cannot be fully hidden despite provision in the Ada language to remove such dependency.
- Code had to be changed because of the lack of standardization of the Ada language in some area. For example there is no standard package of mathematical functions.

- In an hosted environment the implementation of interrupts (LRM 13.5.1) is usually not supported and is sometimes replaced by an operating system dependent construct (such as the pragma AST_ENTRY under VMS with the DEC/Ada compiler). To increase efficiency, implementations have to use this construct when it is available. Part of the OASIS code had to be redesigned because of this issue.
- The lack of standardization of the pragma INTERFACE is also the cause of some code modifications.

1.3 Off-line Guidance Requirements (Univ. of Michigan)

The objective of this effort was to determine the system requirements for off-line guidance of a remote technician.

Experiment Description and Summary Conclusion

We investigated the required modus operandi of a remote coaching system while in the off-line state. This was accomplished through a collaboration between the project design team and select personnel from the space physics laboratory. Biweekly meetings were held to review the project design, to highlight any apparent limitations, and to suggest useful additions to the system.

Issues Investigated

What is the level of expertise of the user of the system?

What methods of guidance should be provided?

What are the most effective man/machine communication methods that can be employed?

Experiment Hypothesis

It was hypothesized that the level of expertise, and hence the training time, for a competent technician could be significantly reduced by providing an expert system with which he could interact. The technician is assumed not to possess all of the knowledge necessary to install and maintain sensitive instrumentation.

Method of Investigation

An expert systems programmer from the Robotics Laboratory and a research associate from the Space Physics Laboratory worked together to determine the requirements for off-line guidance. Knowledge concerning the Fabry-Perot interferometer was obtained from the Space Physics Lab research and converted into a set of rules for the expert system by the expert systems programmer. The two met biweekly to discuss the operation of the system, to determine what rules were incorrectly coded, and to evaluate the utility of the current system.

Experiment Results

It was determined that three modes of operation should be provided by the system: inquiry, maintenance and diagnostic.

A novice user of the system uses the inquiry mode as a learning tool to provide a definition of terms used by the expert system and the physical appearance of different components of the instrumentation.

The use of the inquiry mode diminishes as the technician becomes more experienced. The operator can make general inquiries about the operation of the remote coaching system, the specific instrumentation being maintained (A Fabry-Perot Interferometer in this case) and the procedures used for maintenance.

The maintenance mode is used after it has been determined that a specific component of the system is not working correctly. It leads the technician through a sequence of steps which are required to determine the cause of the problem. Each of the steps in the maintenance mode requires a number of substeps. Associated with each substep is an alignment rule which requests the technician to proceed with some action. After this action is completed, and dependent upon the request made by the technician, the system either continues to the next logical step in the maintenance procedure or establishes a communication link to the real expert at the local site. To enhance the communications between the expert system and the technician, a picture of the device related to the current step is displayed on the video monitor. Text is displayed to inform the operator of the required actions and, finally, the system continues to the next step of the maintenance procedure.

The diagnostic mode is used to determine the cause of a problem. The diagnosis mode uses backwards chaining from the initial knowledge that the interferometer is not working correctly to determine the cause of the fault. As an example, the computer system, pressure system, mechanical alignment system, aperture system and laser system must all be operational for a Fabry-Perot Interferometer to work correctly. First the computer system is checked, then the pressure system and so on. If one of these subsystems is found to be faulty, then the conditions for it to be operational are checked. For example, if the Aperture System is faulty then the Input Aperture is checked to see if it is too large. This process is continued until every known cause of a problem has been checked.

The system was programmed using the CLIPS expert systems programming language[1]. It is designed for efficient forward chaining inference and incorporates the Rete Algorithm[3] for high speed pattern matching. The ability for the programmer to define new functions for the expert system greatly enhances the utility of the CLIPS language. The retrieve-image-rules rule which is used to display images from the video database is an example of the use of such functions.

1.4 On-line Guidance (Univ. of Michigan)

The utility of any expert system is limited by the amount of knowledge contained in the system. In anticipation of this lack of knowledge, we investigated methods by which the technician could place a call to the real expert for additional guidance.

Experiment Description and Summary Conclusion

An investigation into the methods which could be used for collaboration between the local expert and the remote technician were investigated.

Issue Investigated

What are effective methods of communications between the local expert and the remote technician.

Experiment Hypothesis

It was hypothesized that a technician will eventually require additional knowledge which is lacking from the expert system and that an effective means of communication with the expert is via normal telephone lines.

Method of Investigation

Several meetings were held to discuss the importance of different mechanisms for communication.

Experiment Results

It was decided that if the remote technician could not correct a problem with the knowledge provided by the expert system, then a mechanism should be provided to enable him to contact the expert for help. For this reason, during all three modes of operation, inquiry, maintenance, and diagnosis, the technician can ask for on-line guidance from the real expert at the local site. The system should then dial up the host computer of the local expert and the technician would log onto the computer in the normal manner. He can then determine if the expert is currently logged onto the computer and if not, he has the option of leaving a message for the expert using the expert's host computer electronic mail system. On the other hand, if the real expert is logged in, a dialog can be initiated between the real expert and the technician. This is done by the remote technician by invoking a communications program on the experts host computer. The expert must then proceed to a workstation equipped with a video monitor and start-up his communications program.

At this point, both the experts and the technicians screens are divided into three windows. Both screens at the remote site and the local site are functionally the same. The window at the top of the screen displays the transcript of the expert system running at the remote site. The two windows at the lower part of the screen are used to display the text dialog between the technician and the real expert. When the dialog begins, both the remote

technician and the local expert can enter text through the dialog windows. In addition both can enter responses to the expert system program running at the remote site, display images from the video database, and capture live images.

Identical video images of instruments and expected sensor outputs are stored in databases at both the remote and local sites. When the technician explains a problem in terms of an object or a sensor output, he can simply display the related image from the video database at both sites. The objective of the design is to provide the expert with a view of the current state of the remote system and to provide effective methods of communicating with the remote technician.

A pointing device overlaid on the video monitor would significantly enhance the operation of the system. Both the real expert and the technician would have access to such a device. An example is a laser power supply containing several control knobs. If the expert wants the technician to turn a particular knob, he can simply point to it. Similarly, we have found that video images of devices are often cluttered with stray components, not immediately of concern. We have temporarily solved this problem by simply having someone point to the particular device with his finger before the picture is taken. The problem with this approach, is the requirement for many pictures of the same scene. A better solution would be to associate a position of the pointer with different rules when a picture is displayed in addition to the name of the picture. Thus, a picture containing several objects could be used by a number of rules.

1.5 Extended Software Control System (UC Berkeley)

In the Space Station era, it is expected that experiments will be built by a consortium of experimenters located in various corners of the world and interacting over computer networks. To facilitate the design, fabrication, and flight operation processes, it will be necessary to share software and data files between various sites. In order to prepare for such scenario, we have extended a software control system (SCS) we developed for flight and data analysis software for the EUVE satellite. Software for the EUVE project is being developed by a team of programmers, scientists and technicians over a local area network connecting a number of workstations. The mechanism has been in use for the past five years and proved extremely beneficial in the hardware/software development for the EUVE project.

The TTPP experiment we undertook was to extend the SCS to include a wide area network (WAN) connecting the Space Sciences Laboratory (SSL) of UCB to a group at the Massachusetts Institute of Technology (MIT). It is aimed for use by the X-ray Timing Explorer (XTE) project, where a portion of the flight software is being developed jointly by the two groups.

Experiment Description and Summary

The EUVE SCS provides an environment in which the software developers and users can co-exist without impeding the progress of either. It compartmentalizes software into three categories:

- New: The software in this category is under active development and the only allowed users of this partition are the software developers themselves. Once the software is fully developed, it is promoted to an area called test.
- Test: Software in this category are used by the software developers and users for testing and validation purpose. The majority of the interaction between the software engineers, programmers and the scientists take place here. After a reasonable amount of testing the software is promoted to the operational area.
- Operational: After the software has undergone a thorough testing it is promoted classified as operational. At this point the software is considered rugged and the users are encouraged to use the codes residing in this area.

Furthermore, it allows critical software and data files to be copied to multiple selected computers on the network, providing backup functions. This allows the detector development group, for example, to continue to gather data and analyze them even if the computer which has the master copy of the software unavailable.

This system has proved to be extremely useful for the EUVE project. Under the TTPP program we decided to test its performance over a wide area network connecting the MIT and Berkeley. We used the internet for this purpose. Appropriate servers were installed on two machines, one at each locations. Various issues relating to file updates and network performances were investigated.

Issues Investigated

The aim of this experiment was to install a software system so that programmers at MIT and Berkeley can jointly develop software for use on the XTE project. Since the basic system was already operational at UCB, the key issues investigated were the bandwidth, delay and reliability of the internet to support the SCS.

Experiment Hypothesis

The two hypotheses for this teledesign experiment are as follows:

1. The technology is mature enough that software development from widely distributed geographic locations using the existing network is feasible.
2. The internet bandwidth and performance are sufficient to support the required operations

Method of Investigation

We had two meetings with Dr. Hale Bradt and colleagues at MIT to discuss the implementation plans. Afterwards, we installed appropriate software at the XTE1 machine at the MIT Center for Space Research. Several files were created at both locations. We

wanted to see how changes at one locations get updated at the other. Depending upon the originating location, the files in one location was designated as the "master" with a "slave" copy at the other location. Every night the SCS updated the changes the files at the slave locations using the master copy. The first attempt included updating the complete file whenever it is changed. This is easiest to implement but required a large data volume to be transferred over the internet. In the final phase, we would include the updates of only those lines of files containing changes. Due to various problems we were unable to complete this version.

Experiment Results

We have successfully implemented the first version of the wide area SCS. We discovered a serious problem with the internet throughput while whole files were being copied. To minimize the amount of data volume over the network, we attempted to implement the second version. Limitations in the internet routing software prevented us from taking advantage of high speed NASA science Internet (NSI) links. During the day, a simple login to MIT from Berkeley was virtually impossible. The situation has improved somewhat since the NSI backbone was upgraded to T1. Late at night the situation was somewhat improved. We concluded that unless the throughput of the internet is increased, this scheme, although very useful, may not be implemented to the satisfaction of the scientific users.

1.6 Software Emulation of 8085 Microprocessors (UC Berkeley)

The second teledesign experiment performed by UC Berkeley simulated the operations of multiple Intel 8085 microprocessors to be flown in the EUVE satellite. The advantages of a microprocessor level simulation of the EUVE instruments are twofold. First, it allows us to test the ISW in ways not otherwise possible before the instrument is complete. Second, provides an interactive debugging environment, not available with the flight development hardware itself.

Experiment Description and Summary

The EUVE uses an integrated hardware/software development strategy. Its implementation uses an End-to-End system which simulates all major subsystems [Marchant et al., 1987; Chakrabarti et al., 1988a]. This simulation attempts to provide the user with one interface with the instrument and software during all phases of the instrument development, calibration and flight operations. Our simulation experiment was to enhance the EES to include the interactions of the microprocessors at a low level. This allows a thorough testing of the software written for use by the flight microprocessors.

Issues Investigated

The EUVE instrument presents a complex microprocessor environment, with eight 8085 cpu's controlling the different instruments and interacting closely with each other to collect data and generate telemetry. As stated above, a software simulation of the microprocessor environment would allow the instrument software to be developed and tested in advance of the availability of the actual instrument hardware. Such a simulation must

duplicate the close interaction of the microprocessors while providing an interactive debugging environment for the programmer. At the same time, the simulation should be fast enough to be able to run at a significant fraction of the actual hardware speed.

Experiment Hypothesis

Multitasking techniques are well-established, and indeed UNIX provides facilities for writing multitasking applications. Such general-purpose facilities come with considerable overhead, however. For this experiment, we hypothesized that a cooperative multitasking scheme, tailored to the needs of the simulator would minimize the overhead associated with task-switching. This is essential for a simulator that must be able to switch between simulated processors at the level of the processor clock speed.

Using a time-slice of a single (simulated) processor cycle would still bog down the simulator with task-switching overhead. It turns out, however, that the events that generate interactions between the processors (detected photons, telemetry interrupts) occur at a much slower rate than the processor clock (a few KHz compared to 3 MHz). Therefore, we also hypothesized that a variable time-slice would allow us to approximate the behavior of the actual instrument processors while enabling the simulator to run at a reasonable speed. The time-slice would be set to a few hundred processor cycles until an interrupting event occurred, when it would be reduced to a single cycle until the event had been processed.

Method of Investigation

The first phase of the simulation used an interpretive programming language called Magic/L. Magic/L is an interactive environment; it was chosen for the 8085 emulator because it provides a ready command interface for the required debugging support. We have verified the operation of the emulator in a multiprocessing mode through the use of two test programs operating in parallel and sharing data through simulated I/O ports.

In the second phase the core of the emulator was rewritten in Motorola 68000 assembly language for use on our telescience Sun Microsystems workstations. (The original version of the assembly language emulator was written by Mr. Steve Rosenthal of MIT). The user interface was maintained in Magic/L.

Experiment Results

The two programs were run successfully on both versions of the emulation. However, the interpretive nature of the Magic/L language imposes severe speed limitations. We found that a single processor simulation runs at a speed that is approximately six hundred times slower than the actual hardware.

The assembly language version was a significant improvement over the Magic/L version. The single processor speed was found to be 33% of the real-time speed. With the additional overhead of the hardware and multiple-processor simulations, the actual speed is approximately an order of magnitude slower than the hardware. A multiprocessor simulation running at, say, 1% of real-time speed, would simulate several minutes of actual operation in an overnight run.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Section 2

Teleoperations Experiments

2.1 Technology for Teleoperation (Univ. of Arizona)

This experiment investigated the issues involved in development of a generic technology for teleoperation of scientific experiments on Space Station.

Experiment Description and Summary

We implemented two seemingly quite different testbed demonstrations. The first involves teleoperation of a forerunner of the Astrometric Telescope Facility (ATF) to be attached to Space Station. The second involves systems and software for Remote Fluid Handling (RFH) in support of the microgravity and life sciences. These were selected in part so that a generic set of technologies for teleoperation could be investigated.

Issues Investigated

- The design of a set of tools which allow teleoperation of scientific experiments. These tools include the human/computer interface at the remote commanding computer, the computer/computer interface between the remote commanding computer and the local controlling computer (intermediate language and communication protocols), and the computer/instrument interface between the local controlling computer and the equipment (robots, telescopes, measurement instruments, analytical instruments, etc.) which may comprise any specific experiment.
- Ways to ensure that these tools are GENERIC and MODULAR, so that they can easily be applied to a wide variety of scientific applications and so that the individual modules can be revised without significant impact on the remaining parts of the software.
- Evaluation of the technologies underlying the above developments and development of recommendations (specifications) for future development.

Experiment Hypothesis

It was hypothesized that the human/computer interface and computer/computer interface modules can be designed to be capable of controlling a variety of diverse experiments without the need for new software development for each application and that only the computer/instrument interface must be specific.

Method of Investigation

The two scientific experiments, teleoperation of an astrometric telescope and teleoperation of a fluid handling laboratory, are described elsewhere in this report. The equipment to be controlled and the scientific data and telemetry required are quite different, but the same hardware/software architecture was used for both.

The architecture consists of a remote commanding computer (RCC) which communicates at 9600 baud over dialup phone lines, Sytek network, or Ethernet network with a local controlling computer (LCC). In both cases the RCC was a microVAX II GPX workstation which houses a human/computer interface consisting of an application of the Operations And Science Instrument System (OASIS) previously developed by the University of Colorado. For the telescope, the LCC is a second microVAX workstation which also runs the telescope simulation. For the fluid handling laboratory, the LCC is a PC compatible with 640K memory, an 8087 co-processor, a 20 Mbyte hard disc, and a LabTender multifunction board. For both experiments, intermediate command language statements, telemetry, and scientific data are exchanged using DECnet and CCSDS packets as the communication protocols. The communications system design is documented in technical report TSL-19/88, "Communications Software Design for Telescience Demonstrations," dated November 1988.

Our testbeds represent the architecture planned for Space Station to a limited extent. The OASIS Workstation represents the Telescience Workstation fairly well. Also problems with packetizing/depacketizing and routing of data from and to the Telescience user can be well studied using our current distributed testbed configuration. Future extensions of our testbeds will allow study of problems with multiple remote observers and also with multiple simultaneously controlled experiments. However, our testbeds cannot and will not answer questions about overall communication and processing delay times incurred as a result of the rather complex communication path between the experiment and his end user. For this purpose, an entirely different testbed is necessary, namely an end-to-end integrated real-time discrete event simulation of the overall SSIS and telescience extensions. Such a simulation does currently not exist in complete form.

Ultimately, we would like to use free-syntax plain English for expressing user directives. We are currently far from this ultimate goal. However, OASIS presents us with an excellent compromise. While the number of directives that can be understood by OASIS is fairly limited in every application, OASIS supports the creation of user interfaces for new applications in a very convenient manner. The user interface is entirely data driven (through an application database), and it takes an experienced programmer a fairly short time to create a database for a new application including the design of appropriate windows and icons (depending on the complexity of the application between one and four weeks).

Thus, while OASIS has been designed to interact with experiments directly, its true strength lies in its user interface capabilities. The current software is a little slow in performing these actions due to the fact that the application interface is completely data driven. Speed and flexibility are always in strong competition with each other. We suggest that it would be a good idea for the University of Colorado to look into the possibilities of creating a data compiler that can be used to compile a new application database into a set of Ada procedures that can be linked with the OASIS software once the new application database has been completely debugged and tested. This could speed up the execution of OASIS by a factor of 10 to 100. In this way, the best of both worlds (flexibility of the data driven approach, and execution speed of the code driven approach) can be combined in an optimal manner at the expense of one (very slow but insignificant) process of compilation and linkage.

OASIS can perform a limited set of tasks related to the LCC, but the software is not particularly strong in this respect. For instance, it seems obvious that we may wish to initiate several actions simultaneously as long as they do not conflict with each other. OASIS allows processing only one command at any given time. Moreover, it is not useful to combine all the activities in one program since, in reality, the Space Station software will be distributed between the end user's Workstation (on the ground), and his experiment (aboard Space Station) with an uplink and considerable onboard processing in between. It would be useful to modularize OASIS for distributed processing.

In our testbeds, we ignored OASIS' capabilities to act as an LCC, and split the task between two separate computers, the RCC which runs OASIS, and the LCC which runs our own Ada-coded programs. This approach allowed us to use OASIS for what it can really do well, namely communicate with the user, and it allowed us to study problems that relate to the distributed nature of the overall SSIS/telescience hardware/software architecture.

Experimental Results

This experiment was extremely successful. The goal of developing generic modular software for teleoperations was attained. It should be noted, however, that this result would not have been possible without the prior development of OASIS by the University of Colorado (LASP). Our conclusions, recommendations, and suggestions for future work are as follows:

a. SCIENTISTS WILL REQUIRE REMOTE MULTIPLE ACCESS FOR TELEOPERATION

While a number of TTPP participants have looked into the problem of remote multiple access to distributed databases, little has been done with respect to multiple control of experiments. Outside of the telescience community, there exists even less awareness of the need for multiple simultaneous users of the same equipment. We have performed a feasibility study which indicates that multiple simultaneous control and/or observation of an ongoing experiment can be attained.

b. THE NEED FOR MULTIPLE ACCESS MUST BE ADDRESSED NOW

It is necessary to foresee multiple simultaneous controllers/observers right from the beginning since this mode of operation calls for additional communication protocols and equipment. For example, if we want to allow an observer to obtain on his screen a shadow image of everything that is displayed on the main experimenter's console, all screen commands, including the house-keeping commands such as changing the background color of a screen window, must be communicatable rather than being treated as strictly local activities. A more obvious example is the use of token-passing (key ownership) to ensure that only one location is in active control at any particular time. It is therefore not feasible that the system be developed for single users only, while the design of a multiple user capability, for financial and scheduling reasons, is postponed until later. It is important that this capability be considered right away.

c. MULTIPLE EXPERIMENTS MUST ALSO BE CONSIDERED

Due to a lack of appropriate coordinating information, all activities are currently treated as independent units. The general scenario is that of an experiment on one end remotely controlled by an experimenter on the other end. In reality though, all communication to and from Space Station will be routed through a central (though distributed) operating environment, the Operation Management System (OMS). The integration of the telescience activities into the OMS will create some problems (such as additional time delay), and will call for additional communication protocols. It is important that the implications of this integration into the OMS be considered early on. It might be useful to create an OMS simulator, and make this simulator available to the telescience community. The TTPP participants will then be able to route their commands through the OMS simulator to more accurately determine what will happen to their experiments aboard Space Station.

d. OVERALL SYSTEM RESPONSE TIMES MUST BE SPECIFIED

We have studied the effect of communication delays on stability of remote closed loop control. It appears that a total delay of 1 sec can be made acceptable by use of special techniques, but much more than that will result in unstable behavior of the control circuit. For reference, the TDRSS hop alone introduces a communication round trip delay of approximately 0.4 sec. Since we will not be able to guarantee 1 sec delay for normal operation, it will not be possible to routinely teleoperate any robots with a remote man-in-the-loop control configuration. Instead, the local robot control must be automated. The remote operator cannot control the robot directly. His normal mode of interaction must be limited to specifying set points and starting the execution of experiment segments using preprogrammed procedures. If something goes wrong, however, it must be possible to fix the problem remotely. For that purpose, it should be possible to request a special communication path which guarantees a delay, including processing time, of LESS THAN 1 SECOND round trip. For normal operation (specification of set points or procedures), our testbed has shown that a response time of 5 SECONDS will be sufficient, or rather that the control circuits can be designed to operate under such conditions.

e. DELAY TIMES MUST BE VERIFIED BY SIMULATION

It is important to study the effect of these multiple software and communication layers on overall system response time. Since development of these software tools has been distributed among numerous contractors, no one seems to have a clear picture of the effect of these multiple software systems on overall system response times. It is suggested that an end-to-end real-time performance simulator be created to study this problem.

f. PUBLIC DATA SWITCHING NETWORKS CANNOT BE USED FOR TELEOPERATION

The performance available through unlimited access data switching networks (such as NSN/NSF) won't suffice under any conditions. The large, unpredictable, and variable delay times imposed by these networks will jeopardize any successful teleoperation of equipment aboard Space Station. There are those who claim that this problem will be eliminated by installation of higher capacity networks, but this is not true unless access is limited. As an example of this, the NSFnet backbone was upgraded from 56 Kb/s to 448 Kb/s last July, and it is planned to increase capacity to 1.54 Mb/s in 1989 and 45 Mb/s in the early 1990's. The problem is that usage is also increasing (100% per year) so that the current grade of service is not predicted to improve much. This situation could be altered with provision of a priority (virtual circuit) capability. We had planned to document this conclusion with quantitative measurements made on the NASA Science Network, but it is still not available to us.

g. VIDEO FEEDBACK HAS SEPARATE REQUIREMENTS

Bandwidth AND delay problems occur with the incorporation of video feedback. Special connections with a guaranteed delay time of below 2.5 sec must be established which bypass the OMS and can deliver video feedback directly to the teleoperator. More research is needed to establish appropriate data links, switching control, and data reduction schemes for different applications.

h. ADA IS AN EXCELLENT LANGUAGE CHOICE FOR TELESCIENCE

Ada has proven to be an excellent choice as the standard programming language! In particular, it is a very convenient language for telescience applications. Especially useful are the EXCEPTION HANDLING capabilities which allow separation of the error handling from the processing of correct commands, the INFORMATION HIDING features which allow a new ability to modularize software, and to distribute the coding of subsystems among several team members without fear of side-effects, and finally the MULTI-TASKING CAPABILITY which allows capture of parallel processes in parallel software modules in a very convenient manner.

i. OASIS SHOULD BE MODIFIED AND EXTENDED

We have extensively tested OASIS for the purpose of teleoperation of experiments aboard Space Station. We found that OASIS presents us with a flexible and convenient state-of-the-art user interface to command experiments. However, OASIS is not yet satisfactory with respect to its flexibility and performance for communicating with the equipment located at the other end of the communication link, and for interactions of multiple experimenters with multiple experiments.

It is clear that OASIS forms a very good starting point for a generic teleoperations software, but that it must be significantly modularized and expanded for future applications. A number of specific suggestions are contained in our final report, TSL-021/88, and have been transmitted to Colorado. These improvements are all feasible, and the cost to update OASIS will be significantly less than that of redesign/re-implementation.

2.2 Teleoperation of an Astrometric Telescope (Univ. of Arizona)

Attached to Space Station, there will be installed an Astrometric Telescope Facility (ATF). Its primary objective will be the detection of planetary systems around other stars. It will be remotely operated from the earth. This experiment is a preliminary investigation of the issues involved in teleoperation of the ATF.

Experiment Description and Summary

In the first phase of this telescience project, a forerunner of the ATF (Thaw telescope, Allegheny Observatory, University of Pittsburgh) was simulated on a microVAX workstation. This telescope model was remotely operated by a second microVAX workstation. An observation scenario was developed on the host workstation using the Operations and Science Instrument System (OASIS) software package developed by the University of Colorado. Commands are sent from the host workstation to the telescope workstation. Various real-time activities of the emulated telescope and data are telemetered back to the host workstation from the emulated telescope for display.

Issues Investigated

The aim of this research is to analyze human/computer communication interfaces that are convenient for scientists who will remotely operate their instruments, as well as the computer system architecture and communications infrastructure which supports these interfaces. It is also the first step toward control of the Thaw telescope, which may be accomplished during the next phase of this project (see the Future Work section). The final goal is the development of a very realistic and accurate testbed for prototype simulation of the ATF itself. Telescience aspects to be investigated include teledesign, teleintegration, telecalibration, teleoperation, teleanalysis, and telemaintenance.

Experiment Hypothesis

It was hypothesized that safe effective robust teleoperation of an astrometric telescope can be accomplished if (and only if) sufficient sensors, controls, and safeguards are carefully incorporated in the original design. It was further hypothesized that a data rate of 9600 baud would suffice for all communication needs, including return of scientific data, provided that a suitable video data compression algorithm could be developed.

Method of Investigation

Collaborators on this experiment included the Lunar and Planetary Laboratory of the University of Arizona (Eugene Levy), the Allegheny Observatory of the University of Pittsburgh (George Gatewood, John Stein), and NASA Ames Research Center (Kenji

Nishioka, Robert Jackson). After an agreement had been reached on the experimental goals during a meeting of all parties involved at the University of Arizona, a second meeting was held at Allegheny Observatory to specify a list of controls, safeguards, telemetry functions, and scientific data to be included in the THAW telescope simulation. Several observation scenarios were run on the telescope to get a clear picture of the details of operation and to measure the mechanical time constants of the instrument. All discussions were taped (protocol interview technique) in order to be able to retrieve as much information as possible. This technique also allowed measurement of the mechanical time constants simply by talking.

Based on the above investigations, a preliminary choice of the parameters for the scenario of the astrometric telescope demonstration, including the functional design of the language between the local and remote workstations, was completed. Details of this design are documented in a 60 page technical report, TSL-004/87, "Teleoperation of the Thaw Telescope at the Allegheny Observatory: A Case Study," dated December 8, 1987.

The simulation of the Thaw Telescope was written in Ada, and emulates the functional operation of the telescope in the Observatory. Simulated telemetry and scientific data are sent back to the host workstation for processing and display. The observation scenario residing in the remote commanding computer is an OASIS application tailored to such purpose. In its initial phase, the teleoperation closely reflects the current way of operating the Thaw telescope. However, it is a subsequent goal of this project to experiment with alternative user interfaces in order to determine an optimally convenient way for the scientist to remotely interact with the Space Station ATF. The design of the improved human/computer interface will be done in close interaction with the scientists.

The remote commanding computer (RCC) communicates at 9600 baud over dialup phone lines, Sytek network, or Ethernet network with a second microVAX workstation. Intermediate command language statements, simulated telemetry, and simulated scientific data are exchanged using DECnet and CCSDS packets as the communications protocols. The second workstation includes the functions of a local controlling computer (LCC) as well as the telescope simulation. The telescope simulation includes a failure mode simulator, which is used for training purposes and also to aid in the design of a more robust remote control software. The telescope simulator has been documented in Alfie Lew's MS Thesis entitled: "Astrometric Telescope Simulator for the Design and Development of Telescope Teleoperation". This document is available as Telescience Laboratory Report TSL-016/88.

The communications functional design contains provisions for future addition of multiple remote experimenters (multiple RCC's) running OASIS and one local controlling computer connected through a communication network. The software design will allow one OASIS user, holding an allocatable privilege key, to send intrusive telecommands, such as slew the telescope, or nonintrusive telecommands, such as show telescope position, to the simulator. Other (non-privileged) OASIS users may either issue nonintrusive commands to the local controlling computer in order to observe the ongoing experiment, or may request a shadow image of the privileged user's session to be displayed. Non-key holders may also exchange messages with the privileged user and with each other (e.g. to request a transfer of the privilege key), and they can schedule a privileged session for themselves at some time in the

future. Details of the communication system design are available in technical report TSL-019/88, "Communication Software Design for Telescience Demonstrations," dated December 1988.

Development of this demonstration is complete. Visitors are welcome at any time. A videotape of the demonstration is also available.

Experiment Results

Observations

- Remote operation of ground based telescopes is NOT commonly done, not because the control task is particularly difficult to solve (it is not), but because most telescopes are not properly equipped with the hardware required for remote operation.
- The most serious obstacle to remote operation of a telescope seems to be safety considerations. Most earth-bound telescopes have not been equipped with sufficient safeguards that would allow for robust, error free, and accident free teleoperation of the instrument. Such problems can be avoided with properly designed hardware and software, however, it is much simpler and cheaper if an instrument is designed for teleoperation from the beginning, than to retrofit an existing instrument.
- There is currently no ground based astrometric telescope which is remotely controlled. Most instruments with some sort of remote operation facility are infrared telescopes and the rest are millimeter wave telescopes. In both these cases, the problems are quite different from those that can be expected when operating the ATF.
- Astronomers are not usually interested in data reduction techniques. For understandable reasons, they prefer to store every bit of information available to them. Consequently, little research has been done on data reduction for transmission or storage of star field images. However, the automatic control circuitry needs the star field information also (for the automated finder scope and guider scope control). The control circuits very definitely do not need the total star field information. On the contrary, the task of these circuits can be greatly simplified if the information contained in the star field is reduced to only those pieces that are needed for control purposes.

Achievements

- Our development of the Thaw simulator has resulted in the definition of a set of sensors, controls, and safeguards needed for teleoperation of the Thaw telescope. Whether or not this is ever actually implemented, it has shown that the control, sensory, and safeguard systems of the ATF should be developed with the end user in mind. Successful telescience applications of this instrument call for a set of controls, sensors, and safeguards that should be defined early on, before the instrument has been totally developed.

- The goal of developing generic, modular software has been attained. Further discussion of this is contained in the RFH and TECHNOLOGY sections of this report. This result could not have been achieved without the previous OASIS development by the University of Colorado.
- The work on video data compression for transmission of telescope pointing data has been completed. A form of run length coding, followed by variable word length (Huffman) coding was found to be the best for this application. This compression algorithm provides reduction from 8 bits per pixel to 0.015 bits per pixel for telescope pointing data. It appears that use of this technique will also allow compression of typical scientific observation data (of a general starfield) for storage or transmission by at least a factor of 10, and perhaps as much as 100. This research has been documented in Wendy Tolle Walker's MS Thesis entitled "Video Data Compression for Telescience", and is available upon request as Telescience Laboratory Report TSL-017/88.

Future work

It was originally planned, during phase I, to move the simulation to Allegheny Observatory and operate it from Tucson. Then, during phase II, teleoperation of the Thaw telescope itself was to be accomplished. This is not currently feasible. The design and installation of the physical interface between the local controlling computer and the telescope is in progress, but proceeding very slowly due to lack of funding. Thus remote operation of the actual telescope and scientific instrument will not be possible for at least another year, and may or may not be desirable at that time. This is because the development of the multidetector medusa instrument head is currently unfunded.

Because of the above, and because availability of the NASA Science Network (NSN) has slipped until at least January, it will not be useful now to physically move the telescope simulator to Allegheny and operate it from Tucson. All the objectives of this portion of the demonstration can be accomplished in a more timely and cost effective manner by sending the simulator software electronically to the University of Colorado and demonstrating teleoperation of the simulation from Tucson. This could also be done in the reverse direction; ie. operate the simulation in Tucson from a workstation running the OASIS application in Boulder.

In view of the above developments, it seems most logical to proceed with the design, implementation, and teleoperation of a realistic simulation of a space station astrometric telescope facility for the next phase of the Telescience Program. The effort would begin with development of a concise statement of the objectives of the testbed, based on requirements for autonomous teleoperation of the telescope facility, as well as provisions for telemaintenance, telecalibration, and teledebugging (response to anomalous events and conditions).

A preliminary list of these objectives includes:

- Identification of a complete set of controls, sensors, and safeguards needed for remote operation of the ATF.

- Identification of a complete set of scenarios for normal operation of the ATF, and development of a set of software modules implementing these scenarios.
- Identification of a generic set of rules which detect anomalies in time for corrective actions to be taken, determination of a complete set of exception procedures to identify anomalies once they are detected (shut-down procedures), and finally, development of a complete set of recovery procedures to return to normal operation after the causing anomaly has been removed (start-up procedures).
- Design of an appropriate user interface to remotely operate the ATF in a convenient fashion for calibration, maintenance, and other similar functions.

The refinement of this set of objectives will be based on a detailed list of requirements resulting from discussions with experienced researchers at LPL and Allegheny Observatory to identify probable scenarios. From this list of requirements will be developed a comprehensive list of the activities to be simulated. This process will also involve appropriate NASA centers (probably ARC and JPL). It will then be a relatively straightforward task to extend the existing Thaw simulator into a realistic ATF simulator, and to demonstrate teleoperation of the simulation. Depending on ultimate availability of an automated multihead detector system, it may then be useful to proceed with the teleoperation of the Thaw telescope at some later date.

2.3 Teleoperation of a Fluid Handling Laboratory (Univ. of Arizona)

This experiment investigated the teleoperation of a robot controlled laboratory, such as those which will be installed on Space Station for scientific investigations in the Life Sciences and Microgravity Sciences.

Experiment Description and Summary

The initial demonstration involved teleoperation, with robot assistance, of a laboratory which provides automated handling and analysis of fluids, such as those which might be extracted from laboratory animals or human subjects as part of the life sciences program or sent from earth to be processed as part of the microgravity sciences program. The experiments selected were determination of the pH of a solution, and the separation of a solution into its charged components using electrophoresis.

Issues Investigated

The aim of this research was to analyze human/computer interfaces which are convenient for scientists to remotely operate their instruments, as well as the computer system architecture and communications infrastructure which supports these interfaces. It was also intended to investigate the special hardware required for remote fluid handling under microgravity conditions. Telescience aspects investigated are related to teleoperation and teleanalysis.

Experiment Hypothesis

It was hypothesized that automated remote fluid handling can be implemented using robot assistance, thus saving crew time. It was also hypothesized that video feedback is essential to such operation, and that remote man-in-the-loop control is not feasible due to the communication and processing time delays.

Method of Investigation

Collaborators on this experiment included the Center for Separation Science (a predominantly NASA funded research facility) of the University of Arizona (Milan Bier) and NASA Ames Research Center (Daryl Rasmussen, Frank King).

The experiment was performed using a remotely commanded but locally controlled laboratory robot. While the local control loop is fully automated, the telescience user can monitor and direct his experiment through the remote command loop. The required circuitry for the teleoperation of the electrophoresis instrument and the pH instrument was designed and constructed, and the prototype setup was mounted on two mobile instrument racks in a configuration suitable for access by the robot.

A specialized syringe adapter was also designed and constructed. This device provides accurate positioning of syringes for robot pickup, injection of precise fluid volumes, and disposal of used syringes. Efraim Raize designed the syringe driver assembly and a special interface for the isotachophoresis instrument. This work is documented in our Telescience Laboratory Reports TSL-011 "Computer Interface for Electrophoresis Apparatus," and TSL-012 "Syringe Driver Assembly for Automated Fluid Handling Laboratory."

A case study of the software necessary for the operation of the fluid handling laboratory from a remote site was then completed. This software addressed such issues as the user interface, the communication link between the local and remote sites, the commanding of a robot to perform required tasks, and the utilization of the instruments to perform the requested experiments.

The remote commanding computer is the same microVAX workstation that has been used for the astrometric telescope experiment. The local controlling computer is an IBM PC compatible with 640K memory, an 8087 co-processor, a 20 Mbyte hard disc, and a LabTender multifunction board.

The software interface design consists of three parts: a human/computer high level macrocommand interface (an OASIS application) which allows the user to easily operate the laboratory from a remote location without having to be a fluent programmer, a machine/machine medium level command interface (intermediate language) which contains the set of commands internal to the system and enables the communication between the remote commanding computer (RCC) and the local controlling computer (LCC), and a machine/instrument low level command interface to the laboratory robot and the instrument rack. In this design, high level commands are successively decomposed into lower level commands which finally map into the hardware language of the equipment.

Software modularity was a primary design goal. If the laboratory robot is to be replaced by another type of robot, only the code that maps the intermediate language interface into the machine/instrument interface needs to be modified. If the user surface is replaced by another human/computer interface, such as voice activated command input, only the code that maps the user input into the intermediate language needs to be modified.

The design of the user interface (OASIS application) and the functional design of the other two interfaces are documented in Byron Hack's MS Thesis "Man/Machine, Machine/Machine, and Machine/Instrument Interfaces for Teleoperation of a Fluid Handling Laboratory Aboard Space Station," which is available as Telescience Report TSL-013.

The communication system permits teleoperation by 9600 baud dial up modem, Sytek data communication network, or Ethernet data communication network. It was designed and coded by Richard Bierenz and Jerry Hunter and is documented in TSL-019/88, "Communication Software Design for Telescience Demonstrations," dated November 1988. The OASIS databases were written by YaDung Pan, and the scanner, parser, and command interpreter for the local control computer were written by Alfie Lew. This work is documented in TSL-020/88, "Teleoperations Software for Remote Fluid Handling," dated November 1988.

The actual fluid handling device was then converted to a motorized pipette. It was hoped that the use of this device would eliminate the need for the syringe holder and fixture, and reduce the overall weight of the components used in the system. Since the pipette was designed for use in ground-based laboratory work and picks up fluids by creating a vacuum in a chamber above the disposable tip, changes were necessary to prevent the fluid from floating up the chamber in reduced gravity. This was done by using syringes with needles rather than the provided plastic tips. Each syringe was fitted with a membrane attached to a spring, so that fluid pressure pushes the membrane/spring pair up the syringe barrel, with the motorized pipette providing the driving force. Holding of the pipette by the Scorpion gripper was impractical since the 12 inch length of the pipette and the attached syringe caused unacceptable positioning errors of the needle tip. Therefore, the entire wrist assembly was replaced by another assembly that holds the pipette. This assembly retains the up/down (pitch) motion of the original wrist and eliminates the no longer needed roll motion.

This experiment has been completed. Visitors are welcome to see the demonstration. A video is also available, as is the final report, TSL-021/88.

Experiment Results

Observations

- Analytical work in the life sciences or microgravity sciences cannot proceed in an economically feasible fashion without facilities for automated remote fluid handling. The work is tedious and repetitive, and crew time is exceedingly expensive. Thus it is essential that the telescience principles developed in this testbed be extended and integrated with other activities such as the Space Station Life Sciences Glovebox. In particular, an instrument rack (automated

laboratory) should be immediately added to the design, adjacent to the glovebox, with provision for passing samples back and forth between the two areas.

- The robot configuration must be geometrically compatible with the workspace. For example, in front of a flat rack, it is inappropriate to use a robot that is unable to move horizontally and vertically along the rack. It is insufficient that we are able to reach each point in space somehow. Unless the robot configuration matches the workspace topology, the necessitated indirect approximation of the desired robot motion is paid for by an unavoidable loss in positioning accuracy and by unnecessary complexity of the command sequences.
- The handling of small amounts of fluid in space calls for special equipment (syringes or pipettes). The laboratory robot must be able to handle these devices. Today's commercial laboratory robots are not equipped with an appropriate end effector for that purpose.
- The major fluid handling problems which must be solved relate to contamination control and waste disposal, coupled with the fact that there can be no air/liquid interfaces which are not completely controlled with surface tension.

Achievements

- Our testbed has produced the design of an appropriate end effector/ syringe coupling. Several alternatives were investigated, and several alternatives were actually built. This approach works, but is somewhat cumbersome in terms of detailed steps required in the robot programming, and in the amount of waste produced.
- We are currently experimenting with a setup using a motorized pipette. In this approach, only the cheap, light, and small plastic pipette tip needs to be discarded. All other parts are reusable. It is necessary to modify the tip to provide containment/insertion for contamination control, and to avoid air/liquid interfaces.
- Successful teleoperation of a representative (albeit small scale) fluid handling laboratory has been demonstrated. We have been successful in producing generic human/computer and computer/computer interfaces which work equally well for teleoperation of two quite different demonstrations. This could not have been accomplished without the prior development of the OASIS software by the University of Colorado.
- It was established that color video feedback is absolutely required for teleoperation of a robot controlled fluid handling laboratory. We have determined that the recommended video data compression technique for the robot/laboratory observation data is one that currently has wide-spread use in the area of videoconferencing. Since color images are needed for this application, data rates of 80 to 400 kbits/sec (depending on rates of motion) are

required for adequate image quality. The results of this study are documented in Wendy Walker's MS Thesis "Video Data Compression for Telescience" which is available as technical report TSL-017/88.

- It was determined that man-in-the-loop remote control of the robot will not work satisfactorily if the round trip delay time (consisting of communication delays and processing delays for both the uplink and the downlink) is longer than approximately one second. Under normal operation, such a short delay time cannot be maintained. The current thinking in the Space Station community is such that a round trip delay time of five seconds can be reasonably expected, but end-to-end integration real-time performance simulation of the Space Station Information system (SSIS) including the communication links (through the TDRS satellite) must be used to verify this unproven hypothesis. It was determined that, under normal circumstances, a delay time of five seconds will suffice if the robot is operated with local automatic control circuitry, whereby the remote commander is limited to determining the set points. That is, the remote commander can tell the robot to extract 0.5 ml of liquid from container B, but cannot step the robot manually to the desired position. It is suggested that NASA provide emergency channels that can be accessed for a limited time period after something went wrong, for example to get a jammed robot back on track. These emergency channels should guarantee a round trip delay time of less than one second. The results of this study were published in the MS Thesis of YaDung Pan entitled "Teleoperation of Mechanical Manipulators Aboard the U.S. Space Station" which is available upon request as our Telescience Laboratory Report TSL-002/87.
- The varying and unpredictable time delays associated with public packet switched communication channels (for example using the NSF backbone) will jeopardize any reliable teleoperation of robots aboard Space Station from the ground since it is very unlikely that even the five seconds time delay that we talked about above can be guaranteed under this mode of operation.

Future work

While it is important to design the robot, as outlined above, in an intelligent and flexible manner, it is generally cheaper and better to adapt exotic instruments to the capabilities of the once designed robot, than to retrofit new capabilities into the robot. We therefore suggest that an appropriate laboratory robot be designed first. An important component of this will be the design of generic end effectors. In particular some research is needed to develop a more flexible and dextrous hand. Thereafter, all instruments which are to be placed in the instrument rack should be designed such that they can be easily manipulated by the laboratory robot. This will necessitate a major redesign of essentially every single laboratory instrument to be placed in the rack, but they must be redesigned and qualified for space use anyway. While our current testbed helped to develop a set of design parameters for the laboratory robot, we currently do not have the financial means to build a prototype of such a robot. However, in a later stage of the telescience program, this is a task we would be well qualified to pursue.

More research also needs to be done with respect to the type and amount of video-feedback necessary for remote fluid handling aboard Space Station. While it has already been established that video-feedback is an indispensable component of this telescience application, we don't know yet exactly where to place the cameras, or how many of those will be needed.

2.4 Acquisition of Astronomical Images From A Remote Infrared Camera (IR-Camera) (Univ. of Arizona)

The IR-Camera experiment involves the operation of an infrared array camera directly controlled by an IBM PC/AT at the telescope remotely connected to a Sun workstation.

Experiment Description and Summary Conclusion

The IR-Camera experiment objective was to determine the most efficient means to provide remote access to near real time data and real time control of an infrared camera during observation. To provide a realistic model for the observing situation, the TTPP experiment used an existing 64 by 64 element HgCdTe array camera in use at Steward Observatory. This camera is typical in size of the current generation of infrared arrays, and produces astronomical data of roughly the same volume [64x64x2 bytes] expected in anticipated space infrared instruments. Moreover, the space-borne instruments will have similar post-processing requirements to provide usable quick-look images.

The infrared camera's local computer consists of an IBM PC/AT microcomputer running a camera controlling program written in assembler and C. There are a number of limitations in this mode of operation. First, current use of the program requires the astronomer to sit near the telescope dome with the controlling PC/AT which is directly connected to the camera. Second, because the microcomputer is a single-task machine, the astronomer cannot conduct any near real time analysis on the data except between observations. This situation provided us with an opportunity to focus on some of the telescience issues involving teleoperations and teleanalysis and to investigate the enhancements available with more recent workstation technology.

Issue Investigated

Three areas were addressed with this experiment; hardware and communications requirements for real-time remote observations, software requirements to provide the quick-look displays and, the user interface.

Experiment Hypothesis

It was assumed that from the results from this experiment, we could identify the minimum communications requirements to conduct remote real time instrument control for infrared astronomical observations and the processing requirements for near real time analysis of raw data from those observations.

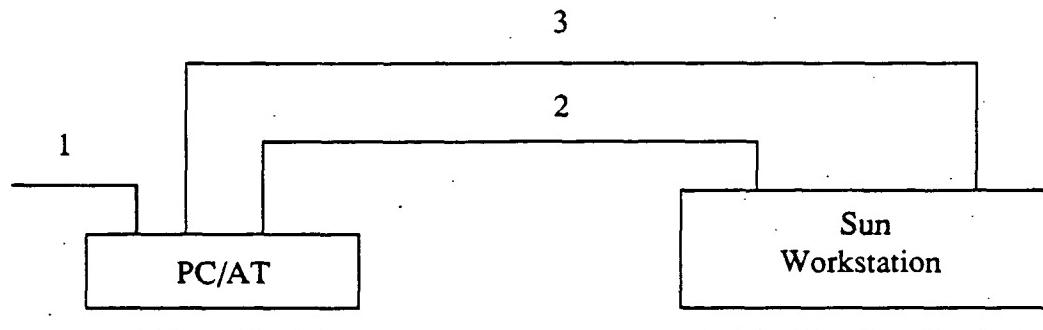
Method of Investigation

Since the infrared array camera had a considerable heritage in hardware and software, as much of this as possible was retained and an extra layer for interfacing with the workstation was added.

Assessing Hardware and Communication Requirements

A number of connection methods between the PC/AT and Sun workstation were considered. Since thickwire ethernet cabling was not available at any of the observation sites, it was assumed that remote connections would be either one of three ways; thin ethernet, serial direct connect or serial dialup. The configuration used for these initial tests is shown in Figure 1. The form of the interconnection was driven by the desire for simple interfacing with the existing PC/AT control program and by the very asymmetrical data flow requirements. Specifically, the data rates necessary from the workstation to the PC/AT for actual control of the camera are extremely low (command streams of a few hundred bytes every few minutes), while the transferring of image data back to the workstation requires much higher rates. In general, the maximum comfortable delay in image display for most users is 10 seconds. For a 64 x 64 frame the required bandwidth is then greater than 1 Kbyte/sec. For the control channel we used a direct serial connection between the Sun and the PC/AT, while the data are shared (using Sun's Network File Server) over a thin-wire Ethernet link.

FIGURE 1
IR-Camera Experiment Configuration



1. Parallel connection to camera interface electronics. PC/AT running camera control program
2. Serial connection between Sun and PC/AT. PC emulation using cty and Kermit.
3. Thin ethernet connection between Sun and PC/AT using Sun PC/NFS.

Assessing Software Requirements

The evaluation process involved identifying existing software that have wide use within the astronomical community. The "de facto" standards chosen include the UNIX operating system, Kermit and TCP/IP for communication protocols, and the IRAF package for image analysis.

Using Sun's Network File Server (PC/NFS), the PC remotely mounts a file system on the remote Sun workstation. This mounted file system is drive "G" to the PC. The camera control program stores an observation frame in it's memory then copies it to disk drive "G". This gives the astronomer using the applications programs running on the Sun immediate access to the last frame observed.

The actual experiment took place at Steward Observatory in Tucson. A simulator program was written that duplicates the actual camera control program except for software stubs to the hardware calls. Requests to perform an observation result in simulated images being stored and appropriate echoing to the observer.

Assessing User Interfaces

We investigated two user interfaces as an environment for the remote operation of the camera, the Transportable Applications Executive Plus (TAE+) and the Image Reduction and Analysis Program (IRAF). A third package that could be suitable for the user interface is OASIS, and experiment control program developed at the University of Colorado. Since a UNIX version of the program was unavailable during the period of the TTPP, we did not consider it at this time.

After trying both packages, it was decided for this experiment to use IRAF as the user environment. Although IRAF is an analysis package rather than an instrument control interface, it contains most of the processing routines needed to convert the raw astronomical data into usable quick-look images. Moreover, IRAF is becoming a standard within the astronomical community, and most anticipated users of this environment will already be familiar with the package.

Experiment Results

In general, we found that using the network file server with IRAF to be an acceptable mode of operating the infrared camera. We were able to control the camera from a single window on the Sun. Because of the limitations inherent in the redirected screen display from the PC/AT, the instrument control interface was relatively crude. However, this was not a serious limitation since the required level of user interaction for the instrument was low. As would be typical of many astronomical situations, the bulk of the user time is spent interpreting the quick- look data. For those tasks, the large collection of analysis functions available under IRAF are quite valuable. The "transfer" of data using PC-NFS was more than adequate for the 64 by 64 images.

The difference in transfer rates for moving the image data in the PC's memory to the AT's hard disk vs. to the mounted file system on the Sun workstation were indistinguishable to the user. For our test frames (64 x 64 x 2 plus 1k header), the transfer time for each frame was less than 1 second. The experiment also used Steward's internal network to see if transfer times would decrease within a multi-user network environment. The transfer times were, again, acceptably short. For the largest currently anticipated arrays for infrared astronomy (256 x 256) a bandwidth of >13 Kbytes/sec will be needed to keep quick look times acceptable to most users.

2.5 Teleoperation of a Ground Observatory (Univ. of Colorado)

LASP has conducted a study of many of the key operations approaches that have been proposed for use in the Space Station era to provide a convenient, flexible, and scientifically useful environment for remote, interactive operation of space science payloads. LASP has implemented the key elements of operations approaches into an Astronomy Testbed at the University of Colorado (CU) using the OASIS teleoperations package, commercial communication services and the 16-inch telescope with it's scientific instruments at the university's Sommers-Bausch Observatory.

Experiment Description and Summary Conclusion

Using scientific instruments and operations concepts that are similar to those now planned for Space Station, this testbed used a ground-based telescope linked to a remote observatory via phone lines and a video network. An interface was developed on OASIS to allow a scientist, located at great distances from the observatory to control the telescope remotely by sending commands to an on-site system and receiving telemetry from the on-site system. This interface was exposed to a diverse group of telescope users. The teleoperations interface was reviewed by each user and modifications were made to the interface. This interaction between interface developers and science observers resulted in an interface that provided the maximum benefit to the scientists.

The astronomy testbed is made up of the following components:

Telescope	16-inch F12 Cassegrain telescope by DFM
Photometer instrument	UBV photo diode photometer by OPTEC-SSP3
CCD Camera instrument	SONY
On-site Controller	APPLE II-E
Commercial Phone Links	2400 baud auto-answer SCHOLAR modems over normal commercial phone lines
Video Network	5 megahertz local area network
Slow Scan Video Analog	video signal at 35 second update
Remote controller	OASIS: Operations and Science Instrument Support teleoperations package executing on a microVAX-II workstation.

The primary conclusions can be summarized as follows:

- A mixture of methods must be used to insure the safety of on-site crew and the safety of the instrument and facility under remote control. This includes both local and remote interlocks and reactive control. The use of command pre-checks and telemetry limits is necessary and must vary as the current operating conditions vary. It is clear that each application must be able to implement and modify these safety issues to meet their particular demands.

- The effects of light-time delays and limited sensory information on the accomplishment of science objectives can be compensated for by goal oriented control sequences using a user interface that includes iconic representation of information and sufficient command protocols between the local and remote controller.
- The remote controller must have access to all local status information on demand or at commandable rates.
- Due to the requirements of feedback control loops some tasks can only be carried out at the local controller but can be initiated and monitored by the remote controller.
- OASIS provides a consistent and effective means for control of a variety of science instruments and platforms. The success of OASIS lies in its ability to implement user requests in a variety of ways, allowing users to tailor their own display icons, color coded displays and modeless menus. There is clearly a need for interfaces that can accommodate both the novice who wants very few choices and the expert who needs complete control over the interface. The flexibility of OASIS allows the interface to be customized on a user by user basis. The OASIS interface was extremely effective for telescience applications since the users had a hand in its design and modification.
- Operations strategies can be defined in advance as procedures for those operations that are repetitive. Menu options can enhance operations by providing the necessary options to define an experiment in situations where scientific observations are defined by targets of opportunity or specific environmental objectives rather than being limited to block schedules. Scientists need to establish procedures with limited options to ensure good results when these opportunities arise.
- The allocation of control, from total user involvement to total automation is a function of the specific task and is highly dependent on the maximum tolerable closed loop delay time.
- Instruments must be designed to provide information on their status and immediate environment to the distributed user.
- The ability to initialize and fail-safe the instrument from the remote location is extremely important.

The testbed also provided some significant general insights into the advantages of the telescience style of operations. According to the users, telescience allowed them to reduce or eliminate some of the difficulties involved in using a general observatory (for example: working outside in the cold and working in the dark). At the same time, telescience preserved the advantages of scientific experimentation at the observatory including direct

control of operations, experimental and observing feedback, and the ability to react to unexpected conditions. These general advantages will be the drivers to help us overcome any technical challenges to the telescience mode of operations.

Issues Investigated

- Investigate the impact of distributing control and monitoring operations of sophisticated scientific instruments to a remote user and identify any problems associated with operating the instruments remotely.
- Enable a large number of scientists and students to experience the scientific benefits of teleoperations.
- Determine how science instrumentation and experiment strategies can be enhanced to take advantage of telescience capabilities.
- Learn how operations can be coordinated to take advantage of unpredicted observing opportunities or problems.

Experiment Hypothesis

A significant number of users can share a common interface that is efficient and useful to a wide range of users. This interface would involve a combination of communication links, including voice, data and video in conjunction with the control, monitoring and interface capabilities of an OASIS package.

Method of Investigation

The Sommers-Bausch telescope with its existing micro processor control has been enhanced to allow for remote control. This involved software and hardware modifications at the telescope site and installation of communication links between the remote site and distributed user sites. An OASIS interface was developed to allow remote users to control and monitor the telescope with a minimum of training. After the interfaces had been tested, a guest observation program allowed for feedback from the scientists for whom the interface was intended. This feedback resulted in refinements to the user interface.

Experiment Results

The experiment demonstrates that OASIS is effective as a remote control user interface system that can be easily modified in response to user needs. OASIS was able to provide a consistent easy-to-use safe and effective environment for a range of remote telescope users. Key to the success of OASIS was its flexibility and the range of tools to satisfy specific users needs.

This implementation raises the issues of distributed control of multiple remote telescopes by several simultaneous and serial users. These issues should be dealt with in a future study.

The OASIS Sommers-Bausch testbed has generated a great deal of interest at the University and will likely be used in a proposed freshman astronomy class at the Boulder Campus. It is also available as a demonstration system at NASA Ames Research Center.

2.6 Teleoperations Using the OASIS Software Package (Univ. of Colorado)

Over the past several years, the University of Colorado group has been developing the Operations and Science Instrument Support (OASIS) teleoperations software package to provide scientists with the ability to monitor and control their own science instruments during assembly, test, integration, and on-orbit experimentation. The OASIS software was used by several TTPP participants, which allowed us to examine the teleoperations requirements and preferences of diverse scientific disciplines.

Experiment Description and Summary Conclusions

The University of Colorado is currently the only university operating a spacecraft for NASA. In our control center for the Solar Mesosphere Explorer (SME), located on the University of Colorado campus, we monitor the spacecraft and its instruments by acquiring and recording telemetry data and by calibrating, checking, and displaying the incoming data in realtime. We control the spacecraft and instruments by transmitting commands and command computer loads. The software that performs these monitor and control functions for SME was developed by University of Colorado and is an extended version of the software used in the Goddard Space Flight Center's Multi-Mission Operations Control Center.

Soon after SME was launched, NASA requested that we use the experience we had gained through SME to develop a new low-cost monitor and control software system that could be used to test and operate future small spacecraft and complex space instruments. The result is the Operations and Science Instrument Support (OASIS) teleoperations package. OASIS provides all of the monitor and control functions found in the SME control center and other NASA control centers, but OASIS can run on relatively inexpensive workstations which allows the scientists who build space instruments and perform space experiments to directly operate their instruments without having to rely on specialized control center hardware, software, and personnel.

The OASIS software, written entirely in the Ada programming language, comprises six tightly integrated subsystems: communications, data handling, display, command, database management, and user interface. The communications subsystem allows OASIS to receive data from and issue commands to science instruments, test equipment, and spacecraft using a variety of communications protocols including RS-232, IEEE-488, NASCOM, and DECNET. The communications subsystem receives incoming data, strips out individual raw data values from the incoming streams, and passes them to the data handling subsystem for further processing. The data handling software translates raw data values into measurements in meaningful engineering units, like volts, and meters and performs checks on the data, alerting users to conditions requiring their attention. For example, OASIS can monitor the temperature of an instrument and alerts users before the situation becomes dangerous to the instrument or spacecraft. If it is important to respond rapidly to a condition,

OASIS can be directed to react automatically. It can, for example, shutdown an instrument upon detection of an overvoltage condition. Data handling passes processed data to the display subsystem which allows users to view the data in realtime in a variety of formats, including alphanumeric text, graphs, maps, schematics, and symbolic representations of analog display devices like dials, gauges, and switches.

Users issue commands by selecting from on-screen menus, by entering commands through the keyboard in an English-like language called CSTOL, or by executing procedures written in CSTOL. The commands are passed to the command subsystem which converts them into the string of 0's and 1's that can be interpreted by the instrument.

Users tailor OASIS for their application by filling in a database that provides information to OASIS on the characteristics of the instrument that is to be controlled and the nature of the processing to be performed on the data returned by the instrument. After the database is developed, the users can write CSTOL procedures to perform tests and implement operational sequences for their instruments.

OASIS can be used by scientists in conjunction with other software they may have for analyzing their data. A common approach is to use OASIS for commanding an instrument, for acquiring the data from the instrument, and for checking and displaying instrument status in realtime. Science data are then transferred to other programs for further analysis.

Issues Investigated

What are the teleoperations requirements of the different scientific disciplines involved in the TTPP, including remote sensing, life sciences, and materials sciences? Can generic teleoperations software like OASIS support users from all these disciplines or will specialized operations software have to be developed by each discipline? Can the space experiments of the various scientific disciplines actually be carried out through teleoperations?

Experiment Hypothesis

By supplying TTPP participants with the OASIS software, and having the software involved in the wide variety of experiments performed by the TTPP investigators, we expected to be able to gauge the degree to which TTPP experiments benefited from teleoperations. We planned to determine actual requirements for teleoperations by examining the way in which the TTPP investigators utilized the OASIS package.

Method of Investigation

The OASIS package was made available to TTPP investigators early in the project so that use of OASIS could be planned for and designed into the TTPP experiments. OASIS software running on VAX/VMS workstations was used for TTPP experiments at Ames Research Center, Purdue University, Stanford University, University of Arizona, and University of Colorado (see the experiment reports from these institutions for details on their investigations). Members of the C.U. group worked with all the TTPP experimenters who were using OASIS, helping them to use the package effectively, monitoring how OASIS was

being utilized in their experiments, and noting the degree to which the package met the needs of the users. Based on early experience, some small changes were made to the OASIS package to better accommodate the TPPP users.

Half of the TPPP investigators were using Sun/UNIX workstations, and as part of the TPPP effort the OASIS package was ported to the Sun environment. Due to long delays in getting delivery of Sun workstations to support this work, the software was not ported in time to be used in the experiments developed by these investigators. Lessons learned in porting the OASIS package to the Sun/UNIX environment are discussed in a separate experiment description.

Experiment Results

- There are many similarities in the teleoperations requirements across science disciplines. All disciplines had general requirements for communicating with scientific instruments and experiment equipment, for monitoring and displaying experiment status data, and for controlling experiments by sending digital messages to instruments and equipment.
- There are no major technological impediments to giving end users located at their home institutions capability to monitor the progress of experiments in space in very near realtime. For current projects like Spacelab, the investigators must be located at a major payload control center to monitor experiment progress. Today's telecommunications capabilities, coupled with software like OASIS that can acquire and display experiment data in realtime, can allow the data to come to the experimenter rather than having the experimenter go to where the data is. Controlling space experiments from a user's home institution, while assuring safety of spacecraft and crew, is also technically achievable although clearly the issues involved in remote commanding are serious and should be important topics of future research.
- Generic teleoperations software like OASIS can economically meet a wide variety of teleoperations needs. There was general agreement that by having a package like OASIS available, users were able to develop their own monitor and control software. Each experimenter that used OASIS found some desired capabilities lacking, but there was a general feeling that new capabilities could be added to a generic package faster and less expensively than by developing new software.
- The flexibility of generic packages like OASIS comes at the expense of performance. OASIS can be adapted to a wide variety of applications by modifying its database tables. However, using the database information to drive processing carries a significant overhead, and some experimenters found the current performance levels of OASIS to be marginal or lower than what they needed. The TPPP investigations helped us develop a better understanding of actual performance needs. Enhanced throughput should be a major goal of future development of OASIS or other similar packages.

- Some specific recommendations from TTPP investigators for enhancements to OASIS are given below. Additional recommendations can be found in the investigators' experiment reports.
 - Make it easier to learn and to use. Tailoring the database tables for a particular application is quicker than writing new software code, but database development is nonetheless slower and more laborious than users liked.
 - Modularize the package so that the software providing generic functions can be augmented or replaced by software to perform the functions faster or in a way more suitable for a particular application. Modularizing the software would also allow OASIS subsystems to run concurrently on separate processors, thus improving performance and system flexibility.
 - Provide more support for multiple geographically-dispersed users who are monitoring and perhaps controlling the same experiment.
 - Enhance the user interface to meet the need of sophisticated experiments like those involving robotic elements.

2.7 Operations Management System (OMS) Study (Univ. of Colorado)

LASP conducted a study of some of the transaction management approaches that have been proposed for use in the Space Station to ensure that the crew, the spacecraft and the instruments are protected against accidental manipulations, erroneous commands, or malfunctions. Transaction management is an enabling technique for effective teleoperations. LASP implemented key elements of the transaction management approach in a Solar Mesosphere Explorer (SME) teleoperations testbed.

Experiment Description and Summary Conclusion

The approach proposed for the Space Station consists of two kinds of transactions between an instrument and the Operations Management System (OMS). The first (Command Interlock) is a mechanism for preventing a payload from going into a configuration that would cause it to exceed its current envelope (by envelope we mean the envelope of the resources allocated to the payload). Two types of command interlocks are considered for the Space Station: Hardware Interlock where the Transaction Management (TM) function controls the physical enabling of the command; and Software Interlock where the instrument management system requests permission prior to executing the command. The second type of transaction (Reactive Control) is a mechanism by which the OMS components at both the instrument and supervisory levels senses that a payload is going outside of its scheduled envelope and requires the payload to correct its operation.

Along with those two types of transactions, a third (Resource Request) is also considered. With this transaction a payload, sensing that it will exceed its scheduled resource envelope, can ask the supervisory-level OMS to for a larger resource allocation.

Those three types of transactions have been implemented using the SME spacecraft as a testbed. The implementation provides a realistic simulation of command interlock. Simulations of reactive control and of resource requesting are somewhat less realistic since the concept of resource envelope does not apply very well to the current SME instrument scheduling system. However the implementation provides a very effective way to protect the SME spacecraft and the SME instruments against erroneous or unscheduled commands from remote users. The implementation proves that the concept of transaction management can be used in a teleoperation environment to protect the crew, the spacecraft and its instruments effectively against erroneous and accidental commands.

Reference: SAIS Payload Operations Management Concept. Draft Version 2.3 (b).
SAIS Working Group Teleoperations Panel. June 16, 1987

Issues Investigated

The main goal of the study is to address the major issues and potential obstacles to teleoperations: that is, can we ensure safe operation of payloads by geographically distributed users and can these users control their instruments without significant command delays caused by command checking on the ground?

Experiment Hypothesis

We hypothesize that safe operation of payloads by geographically distributed users can be obtained by applying a set of transaction mechanisms that allow the OMS to ensure that payloads are staying within their scheduled resource envelopes, therefore assuring safe operation of the payloads.

Method of Investigation

Command Interlocks, Reactive Control and Resource Requests have been implemented as a proof of concept by upgrading an SME teleoperation testbed that was developed under another NASA contract (GSFC, code 520) in 1987 under the name of Telescience Implications on Ground System (TIGS). The TIGS testbed consists of several OASIS systems (from which remotely-located scientists can control and monitor SME instruments), a Data Interchange Facility (DIF) for distribution of the SME data to the OASIS systems, a Simulator that makes SME instruments look like Space Station instruments as far as the format of the downlink and uplink is concerned (CCSDS packet and SFDU format are used). The Simulator interfaces to the regular SME Mission Operation Control Center for command uplink and telemetry acquisition. It has been upgraded to provide for the transactions listed above and to give a certain level of intelligence to the SME instruments. The upgrade is provided by having each instrument in the Simulator run a "sub-OASIS." The "sub-OASIS" performs the TM functions for the instrument: it implements the instrument-level OMS. The "sub-OASIS" can be seen as an OASIS system with all the user interface functions removed.

Implementation of the Supervisory-level OMS.

In our simulation, the supervisory-level OMS functions are provided by a person, usually the Command Controller (CC) in charge of the SME spacecraft. The supervisory-level OMS controls and modifies (via its own "sub-OASIS") the databases used by the instrument-level OMS.

Implementation of the Command Interlock Transactions.

As a first protection against unscheduled commands we have implemented the notion of a command session. The list of the users authorized at one time to command an SME instrument is stored in a database controlled by the supervisory-level OMS. Only authorized users can open a command session. The supervisory-level OMS can close a command session at any time.

All the instrument commands are stored in a database that is controllable by the supervisory-level OMS. This database mainly contains information on the legality or the criticalness of each command. This database is accessed by an instrument software in the Simulator each time the instrument is asked to execute a command. Because it has sole control over this database, the supervisory-level OMS can disable some commands or even remove them from the database at any time. Deactivation of commands is recognized by the instrument software in the Simulator. The supervisory-level OMS can also indicate that a command is interlocked. Recognizing that the command is interlocked, the instrument software asks the supervisory-level OMS for the permission to execute the command. If permission is granted, the command is forwarded to the SME Control Center for uplink to the spacecraft. It is worth noting that this implementation covers the notion of hardware interlocks (commands disabled in or removed from the command database) and the notion of software interlocks. The command database can also be changed by the supervisory-level OMS according to a schedule so that commands can be interlocked only at certain times. This system provides very good protection to the SME spacecraft and its instruments against unscheduled or erroneous commands. It also provides good protection against unwanted users.

Implementation of the Reactive Control Transactions

At any moment, if something appears not to go according to plan (i.e. an instrument is operating outside of its allocated resource envelope), the supervisory-level OMS can ask an instrument to shut down. This is implemented by having the instrument execute in its "sub-OASIS" a pre-canned procedure. In the current implementation this procedure can only shutdown the instrument in the Simulator. But it could be used to send shutdown commands to the instrument on the SME spacecraft or to update the instrument as appropriate for the envelope infraction. After a shutdown is required, all remote commands to the instrument are rejected. Only the supervisory-level OMS can restart the instrument.

Implementation of the Resource Request Transactions

Each "sub-OASIS" provides each instrument with a mechanism by which it can check its ancillary data against high and low limits. When such a limit is reached or exceeded an instrument asks the supervisory-level OMS to grant it more resources. This is done by having the supervisory-level OMS increase (or decrease) the limit that has triggered the resource allocation transaction.

Experiment results

The experiment demonstrates that Command Interlocks and Reactive Control can be effective in protecting spacecraft and payloads against accidental and erroneous commands. One of the conditions of effectiveness is that all instrument software follow the rule of accessing an OMS-controlled command database prior to any command execution.

The implementation raises the issue of how little visibility the remote users have on the actions the OMS is taking. This issue should be dealt with in a future study.

These techniques are currently being used in the SME testbed to allow remote users at both the Marshall Space Flight Center and at Purdue University to directly control instruments on the SME spacecraft without harming the instruments or the spacecraft or interfering with other preplanned operations.

This testbed has demonstrated that the transaction management approach proposed for the Space Station works and supports teleoperation by distributed users.

2.8 Packet Telemetry, Packet Command, and Standard Format Data Units for Science Instrument Operations (Univ. of Colorado)

The University of Colorado group evaluated packet telemetry using actual realtime spacecraft and instrument data. Packet commands were also tested using actual spacecraft commands. The concept of Standard Format Data Units was used to increase the flexibility of telemetry and command packets.

Experiment Description and Summary Conclusions

Spacecraft telemetry and command systems are changing. NASA's space station and other future missions will use packet-oriented telemetry and command systems rather than the time-division multiplexed telemetry systems and centralized command processors used on spacecraft flown during the past several decades.

In a time-division multiplexed (TDM) telemetry system, a processor onboard a spacecraft assembles frames of data containing a fixed number of data words. The spacecraft's telemetry system builds up a frame one word at a time, soliciting the data to be placed into each word from the appropriate subsystem or instrument. The data words from a subsystem or instrument are typically scattered throughout the frame (and often across the frame) and the data streams must be reconstituted on the ground. Typical spacecraft may have five to ten different frame formats. Through pre-flight negotiation, subsystems and instruments are allocated a fixed portion of the words in each frame format. On sophisticated spacecraft like the space shuttle, frames can be redefined to accommodate new payloads, but

this reprogramming can be expensive. Ground processing of TDM telemetry data has a high overhead because of the effort required to reconstitute the various data streams embedded within the frames.

With packet telemetry, there is no central telemetry processor assembling frames word by word. Instead, each subsystem and instrument assembles and then delivers complete packets of data. The format and length of a packet are determined by the subsystem or instrument that sends it. The spacecraft telemetry processor accepts packets and transmits them to the ground, where they are forwarded to their final destination.

Commands sent from the ground to the subsystems and instruments onboard most spacecraft are routed through a central onboard command processor. The spacecraft command processor partially decodes each command to determine its destination. All subsystems and instruments typically use a common/command format, which means that the command interfaces of science instruments are both driven and limited by the design of the spacecraft command processor. Special problems are posed by microprogrammed instruments, since provisions must be made for program and data uploads.

Packet command systems establish the command packet -- rather than the individual command -- as the medium of exchange in command transactions. The spacecraft uplink processor simply routes packets to their destination and need not decode the command information contained in a packet. Thus command packets can contain commands of almost any format, and they can be used to transport microprocessor program and data loads.

There has been substantial work in preparation for packet telemetry and command. The Consultative Committee for Space Data Systems (CCSDS), an international consortium of space agencies, has produced recommendations for a standard structure for telemetry and command packets. NASA's engineers are working to develop efficient onboard telemetry and command packet systems, but no one has examined packets from the end user's perspective. Many space scientists are unaware of packet telemetry and command and what the use of packets may mean for future instruments and data analysis systems. We therefore undertook an experiment where we packetized data produced by the Solar Mesosphere Explorer (SME) spacecraft and delivered the packets to local and remote investigators. We also allowed users to issue packetized commands to the SME instruments. The SME spacecraft has a traditional TDM telemetry system and centralized command processor, and these cannot be changed in flight. Instead, we captured SME telemetry frames as they arrived at the control center in Boulder, and repackaged the data into CCSDS telemetry packets. Packets were distributed to scientists via existing communications networks. We also received command packets from users over the communications networks, stripped out the commands, reformatted them, and forwarded them to the spacecraft.

Issues Investigated

How much effort is involved in designing and implementing telemetry and command packets for typical science instruments? Will the greater flexibility which packets provide be worth the price of implementation? Can packet-oriented telemetry and command capabilities be integrated into existing data handling and analysis systems?

Experiment Hypothesis

The SME science instruments are typical space instruments, and we presumed that the efforts involved in developing and processing packets for SME instruments would provide a meaningful indication of the costs and benefits of using packets for many future space instruments. We further assumed that the effort involved in developing packet handling software for our Operations and Science Instrument Support (OASIS) teleoperations software package would be indicative of the effort involved in developing packet interfaces for future data handling systems.

Method of Investigation

We had previously built a simple software simulator of space station data and communications systems. This space station emulator takes realtime data produced by the SME spacecraft and its instruments, packetizes the data, and transmits the packets to a simulated space station ground processing center called the Data Interchange Facility (DIF). For this experiment, we updated the emulator so that each SME instrument had its own 'instrument control program' which handled telemetry and command packets for the instrument. In essence, an instrument control program served as the communications processor for a packet-oriented instrument. The code we put into each instrument control processor would normally be placed into the instrument itself, so the amount and complexity of code in an instrument control program is an indicator of the coding required for future packet-oriented instruments.

Telemetry data received from the SME were shipped to the spaced station emulator in realtime, and the data from each instrument were placed into separate packets. The data from each SME spacecraft subsystem were also placed into their own packets. There was a type of packet containing a summary of spacecraft and instrument status, and another type of packet containing a summary of Tracking and Data Relay Satellite (TDRS) status. Packets were routed from the station simulator to the DIF simulator using our laboratory's local area network and the Space Physics Analysis Network (SPAN). Users received and displayed incoming telemetry data using the OASIS software. The OASIS software was modified for this experiment to accept CCSDS telemetry packets and to produce CCSDS commands packets.

Each instrument returned both status data and science data. For each instrument, we designed one packet type which contained status data only and another which contained status plus science data. In general, the number of different packet types produced by an instrument will be proportional to the number of major operating modes. There were two major operating modes for each instrument in our test -- engineering mode and science mode -- and our two packet types have a one-to-one correspondence with these modes. The number of packet types produced by an instrument can become very large if the instrument produces several different data streams that can be present in any combination. This combinatorial explosion can result in so many types of packets that the complexity of instrument software and ground processing software becomes excessive. We did not have this problem in our test, but we decided to examine ways to avoid proliferation of packet types by using another concept developed by the CCSDS: Standard Format Data Units. In

the Standard Format Data Unit (SFDU) concept, each major chunk of data -- the chunk is called a data 'object' -- is preceded by a 20-byte label that establishes the presence of an object, and identifies the type and length of SFDU data objects. Engineering mode packets consisted of only a status data object. Scientific mode packets consisted of a status data object and a science data object.

Each SME instrument had its own type of command packet. Commands for each instrument were defined as a string of ASCII characters. Commands were considered to be data objects and were preceded by a SFDU label. Instrument microprocessor loads were not included in this experiment, but they too would have been identified through their own SFDU label.

Experiment Results

- The software needed to produce and receive packets can fit within instrument microprocessors. About two hundred statements of Ada code were required, including code for the extra capabilities we added, like support for SFDU's. This is commensurate with the code volume and complexity required within instruments for operation with current spacecraft command and telemetry systems. Our Ada packages providing CCSDS packet and SFDU data structures and services are available to any interested party.
- Telemetry packets substantially improved our ability to get instrument data to experimenters. Users in our experiment needed only to specify that they wished to receive packets produced by a particular instrument. Once a packet was received, the data were almost directly usable by the scientist. Because the packets require less ground processing than TDM frames, packet telemetry can reduce latencies in getting time-critical data to experimenters.
- It was easy to create telemetry packets that met the needs of the users in this experiment and to modify packets to accommodate changing demands.
- The use of Standard Format Data Units (SFDU's) increases the flexibility inherent in packets, particularly for instruments that produce multiple data streams. SFDU's make it easy to identify and separate out data streams from a packet, and route the streams to processing and analysis programs. The overhead added by SFDU labels can be high, but SFDU's can result in simpler packetizing and depacketizing software and less ground processing time.
- Packet commands increase the flexibility of the command interface to instruments by allowing the instrument designer to develop commands in a format specifically suited to the instrument. Encoding commands in ASCII is useful because the commands are then both human and machine readable. Command packets make it much simpler to send program/data uploads to an instrument microprocessor.

- Developing the software needed to adapt OASIS to packet telemetry and command was easy and packet communications software is actually somewhat simpler than the software needed to acquire and process TDM frames.
- Existing communications networks like the Space Physics Analysis Network (SPAN) are acceptable means of transporting telemetry and command packets. However, to accommodate the large volume of data expected from future missions, NASA will need to greatly expand the capacity of its science support networks.

2.9 MIT/KSC Space Life Sciences Telescience Testbed (MIT)

Experiment Description and Summary

As an element in the Telescience Testbed Pilot Program, the Massachusetts Institute of Technology Man Vehicle Laboratory (MIT/MVL), Payload Systems Inc. (PSI) and the Kennedy Space Center (KSC) Life Sciences Flight Experiments Program have jointly developed a Telescience Life Sciences Testbed in collaboration with MIT's School of Engineering and the advanced academic computing Project ATHENA. Technical details of the Testbed are available in a System Definition Document.

As part of the testbed, two Vestibular Sled experiments were conducted at KSC by two Experiment Payload Specialists (EPSes). The experiments were remotely monitored and controlled by a Principal Investigator (PI) and one assistant (API) at MIT. The two experiments were:

1. a study of ocular torsion produced by Y axis linear acceleration, based on the Spacelab D-1 072 Vestibular experiment performed pre and post flight at KSC.
2. an optokinetic nystagmus (OKN) /linear acceleration interaction Experiment, based on that proposed and recently selected for development for a future Spacelab mission (NASA '84 AO).

These two experiments were meant to simulate actual experiments that might be performed on the Space Station and to be representative of space life sciences experiments in general in their use of crew time and communications resources.

The PI used an Athena Advanced Visual Workstation to view a realtime video picture of the Sled experiment and data associated with sensors on the subject and on the Sled. Video pictures from KSC were transmitted to MIT via the NASAselect video service utilizing Satcom F2R. In addition, the PI had a two-way voice link with the EPSs at KSC who were respectively operating the Sled experiment and acting as the subject.

During the experiment operations, the PI evaluated his/her ability to monitor and control the experiment and perform a remote coaching function when the surveillance video bit rate was limited, and data links and voice protocols were degraded. The experiment evaluated schemes for video bit rate reduction using operator selected changes in frame rate,

resolution, grey scale and color parameters. The PI also evaluated the type of information displays needed to properly interact with the remote experiment, archive data for near real time analysis, and to maintain written, voice, and video logs.

Issues Investigated

The goals of the MIT/KSC testbed project were to:

- Evaluate the methodology for conducting an actual life sciences experiment over real physical distance with voice, video, and data interaction between the experiment and the remotely located investigators.
- Identify Space Station Information System's (SSIS) Science and Applications Information System (SAIS) requirements for experiments requiring real time voice, video, and data interaction between investigators, station crewmembers, and space station operations personnel.
- Develop telescience methodologies whereby students at universities can become directly involved in space station science activities.

This research focused on the role of the PI in performing a "remote coaching function", and investigated the following issues:

- allocation and use of reduced surveillance video bandwidth
- voice protocols for investigator/crewmember interaction
- investigator workstation design for use with integrated video, text, and graphics

Methods of Investigation

The testbed project was designed as "an experiment within an experiment". The Telescience Experiment was the primary experiment but the simulated experimental payload, in this case the U.S. Laboratory Sled at the KSC Baseline Data Collection Facility (BDCF), was the experiment within the Telescience Experiment.

The project was carried out under the "rapid prototyping" philosophy adopted for the USRA/RIACS university based telescience projects: A high fidelity simulation of the Space Station Information System (SSIS) was not attempted; only those aspects believed germane to the hypotheses under examination were simulated, taking technical shortcuts where necessary. No attempt was made to produce hardware or software which could be directly adapted for the SSIS. Rather, the role of the TTPP testbeds was to define - via simulation - the key technical systems, parameters, procedures and concepts which must be considered in the design of the SSIS in order to maximize the science utility of the space station.

The physical configuration of the MIT and KSC facilities is shown in Figures 1-3:

Figure 1

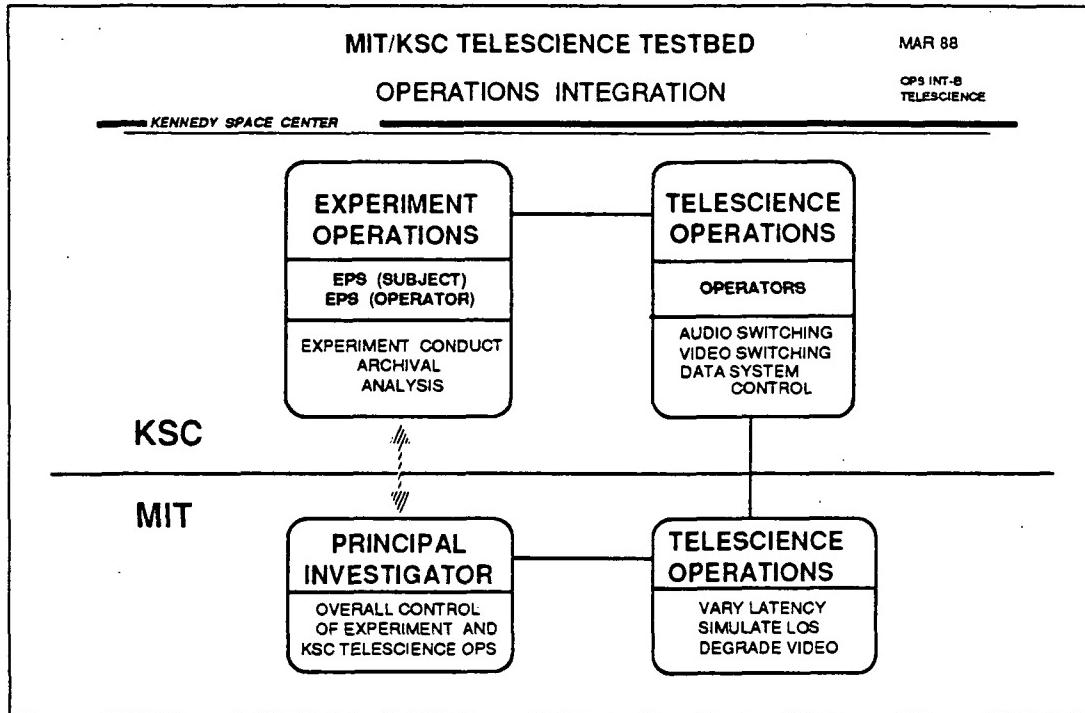


Figure 2

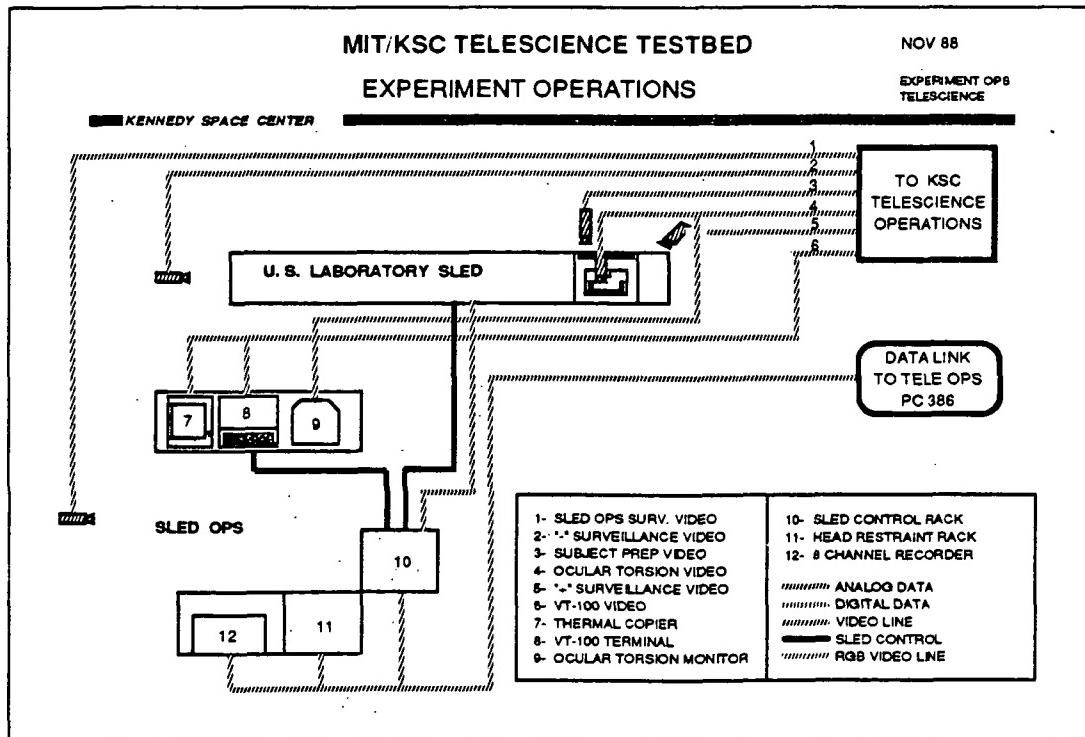
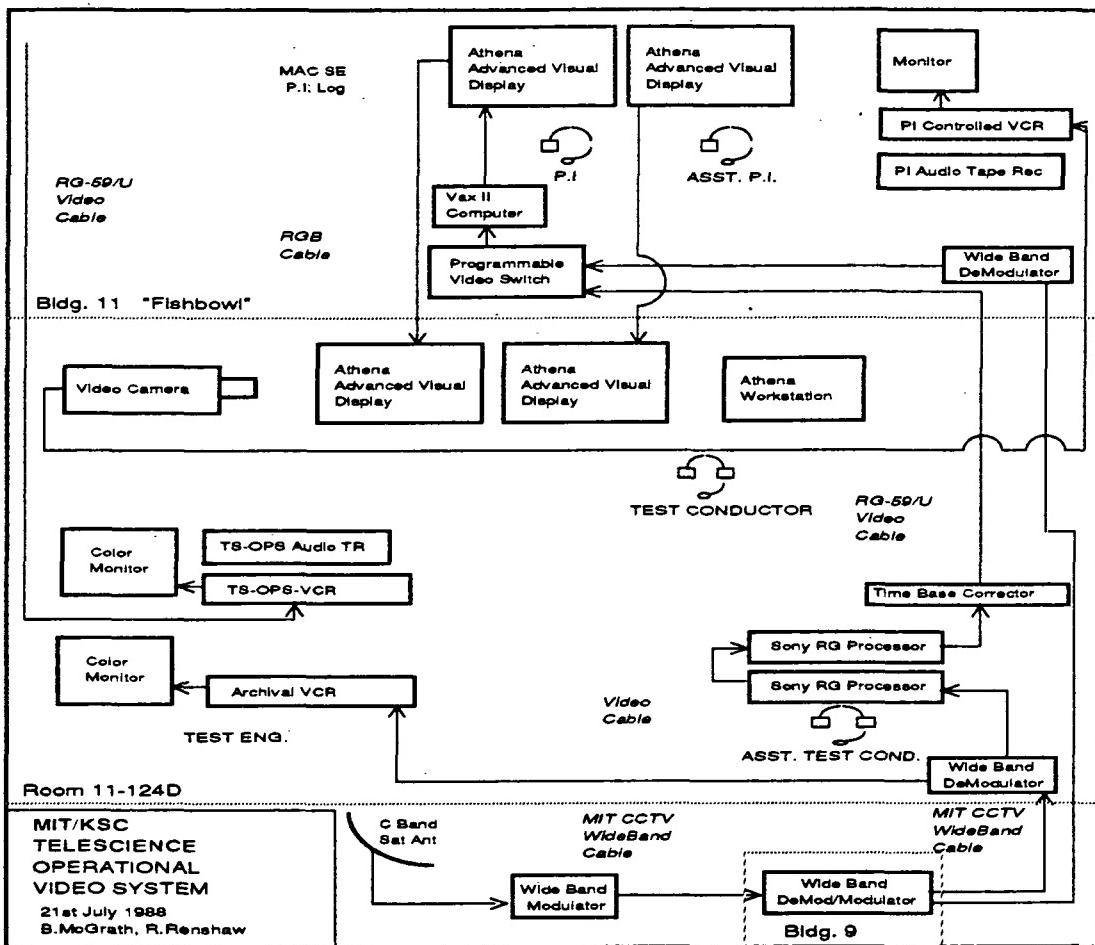


Figure 3



The KSC EPSes were volunteers from the KSC area, selected for previous "operational" experience. None had extensive prior background in space life sciences. The EPSes were trained on KSC generic sled operations procedures in April. In May, they were briefed on the science background of the experiments by the PIs, and then were trained to a approximately a steady state proficiency in experiment operations in June. In July, PI and team training was accomplished and a series of pilot experiments were performed. The formal Telescience Experiment was undertaken for a total period of about 3 weeks in August, 1988, testing 4 evenings/week.

Due to scheduling restrictions on the NASA select video link, actual test sessions began at approximately 5 PM, and were approximately 2 hours in duration, followed by a 1 hour debriefing. During each experiment, the EPSs at the KSC BDCF operated the KSC Sled and performed the ocular torsion or visual vestibular interaction experiment. Their progress was locally documented in detail by a KSC Activity Observer, an individual who was trained to a proficiency level above that of the EPSes and who made detailed notes on the exact time of performance of all timeline steps and crew errors, deliberate or not, and their resolution.

During the experiment, the PI and API monitored and controlled the experiment from the Telescience station in an Athena VAX workstation cluster at MIT. In an adjacent room, the MIT Test Conductor (MIT-TC) systematically varied the voice, video, and data communications link parameters according to a predefined experimental design, coordinating with the Test Conductor at KSC (KSC-TC). Each Test Conductor was assisted by a Test Engineer.

The ability of the PI's to effectively interact with the crew and the experiment as a function of experiment independent variables was assessed.

Independent variables considered included:

- Video bit rate
- Operator control of Frame rate, Resolution, and Grey scale; ("FRG")
- Voice enabled protocol and delay in execution
- Format of PI data display
- PI display capability for data, video, and voice "lookback"

Measurements

The PI was required to keep a standard POCC log of major Sled experiment events and Telescience events such as bandwidth reduction and error detection. The POCC log was entered in a Apple Macintosh computer which automatically time stamped entries. Activities in the area of the PI workstation were recorded by a surveillance video camera mounted nearby.

Assessment of the PI's "monitoring" function during set up and execution was measured using the "error scenario" method: the crew deliberately erred in procedures, and/or hardware and data problems were injected by the Telescience Test Conductor. Problems were solicited from the KSC crew and others familiar with the experiment. The PI was unaware of the contents of this problem set. [This method is basically similar to the "green card" approach used to generate problems in conventional simulations for space missions.] Spontaneous errors also occurred due to EPS errors and data system malfunctions.

Assessment of FRG utilization strategy was by analysis of computer records of FRG choices vs. time for fixed bit rate constraints, and PI log descriptions of the strategy used.

Assessment of team science productivity and PI effectiveness was accomplished using a variety of subjective and objective measurement techniques:

Subjective: Post experiment debriefing questionnaires, using multidimensional subjective rating scales.

Objective: (1) Measurement of number of audio exchanges (basically the number of times the experiment is "held up" to resolve difficulties). (2) Time to complete the experiment. (3) Experiment step number vs elapsed time plots. (4) Quality of PI logs

(frequency, correctness of entries). (5) The number of problems detected and not detected by the PI, and the number of problems solved vs. unsolved. (6) Percentage of sled runs producing usable data.

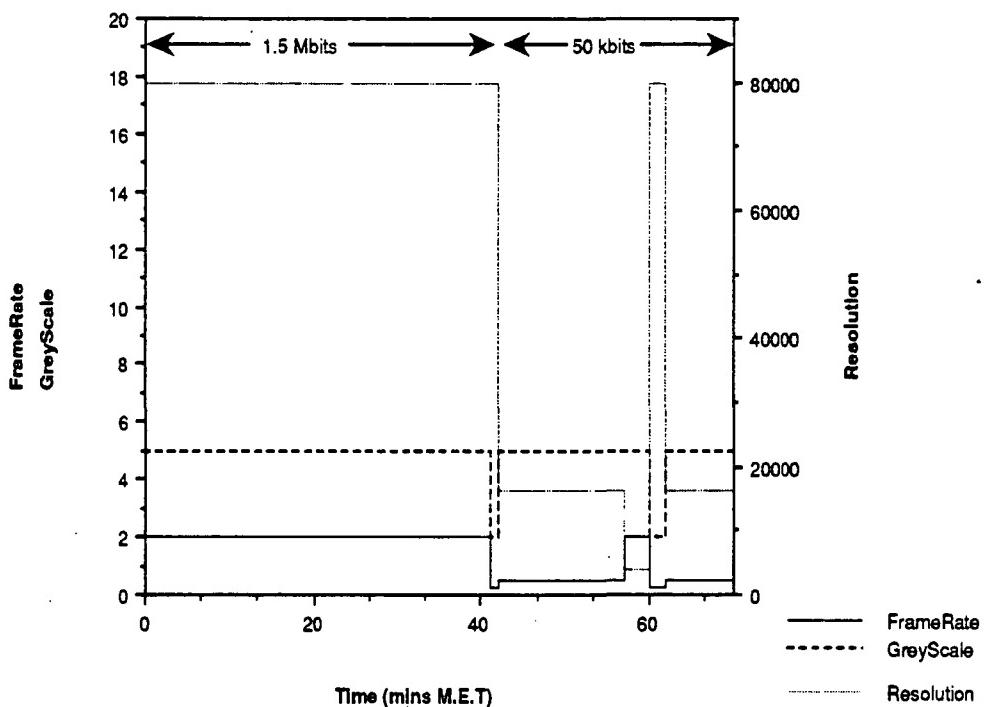
Results

- Nineteen two hour evening test sessions were completed in July and August. These included 11 "pilot" sessions, and 8 "experiment" sessions using a standardized protocol and a statistical behavioral experiment design.
- Real time remote coaching and quick look data assessment in Telescience mode by the PI team clearly were an effective method of detecting and correcting many types of crew procedural errors, and of capitalizing on serendipitous findings. After 7 of the 8 formal experiment sessions, the PIs retrospectively judged that the scientific outcome of each session would have been different in important ways had the PI not been working with the crew in Telescience mode.
- Each of the two PI teams was led by a person (Lichtenberg and Oman) with actual Spacelab mission operations experience. Both PIs felt the specific experiment task demands on the EPSes and PIs corresponded well generically to those associated with actual Spacelab missions. The only major exceptions were in the area of voice communications: On Spacelab, only a single shared air/ground voice loop was available, the crew used "push to talk" microphones, and the crew only reported off nominal or unexpected findings to PIs, usually at the end of the experiment ("negative reporting"). In our Telescience scenario, however, there was only a single experiment underway and there was no Space Station Control Center or Discipline Operations Center cadre, so the crew and PI did not have to contend with other voice communications traffic on the air to ground loop. For technical reasons, our crew at KSC performed the experiment with open mikes. In effect, our scenario emulated a dedicated, open mike voice link capability between the space station crew and the PIs. Both PIs commented that the ability to continuously listen in to open mike conversations between the crew made a major difference in their ability to keep track of the progress of the experiment (both with and without simultaneous video), and were surprised by the difference this one factor made. When PIs could speak to the crew without having to ask for permission, and without the 30 second "voice enabling delay" we imposed in some of our sessions, PIs were much more effective in assisting the crew in troubleshooting, or in following up on unexpected findings. Our experiments demonstrate that remote coaching by voice works. We concluded that dedicated crew-PI voice loops would be of great value on Space Station, and that the crew should where possible perform individual experiments with open mikes. This approach would permit abandonment of the "negative reporting" concept currently used on Spacelab, wherein the crew only comment or initiate PI contact if they encounter off nominal results.

- We gave the PIs independent control of frame rate, resolution, and grey scale ("FRG parameters") subject to an overall surveillance video bit rate restriction, which was systematically varied in different sessions from 12 Mb/sec to 50 Kb/sec. This forced PIs to experiment with different F-R-G parameter trade-offs during the experiment to determine the optimal F-R-G parameter settings for the current situation. A typical example is shown in Figure 4.

Figure 4

VVI 1



Our PI's found that they could "get along" at video bit rates lower than they subjectively "liked". Black and white surveillance video was found acceptable for these experiments. When monitoring the progress of crew activities, PIs found that their choice of F-R-G parameters for surveillance of the crew's activities eventually stabilized, regardless of which experiment step was being monitored. Almost invariably, our PIs were willing to sacrifice frame rate down to the minimum of 0.25 /sec available in order to obtain more than 3 bits of grey scale and the maximum available resolution. (Due to "power of 2" constraints imposed by our image digitization hardware on the choice of F-R-G values available, lower values of framerate were impractical). Due primarily to constraints on the number of test sessions, we were unable to explore a great many combinations, and we did not establish the absolute minimum appropriate frame rate, or establish whether 4 or 5 bits represented the minimum gray scale. The lowest rates used - 50 kB/sec - are well within the digital data

transmission capabilities of the ISDN telephone network or the NASA Science Network, and additional bit rate economies could doubtless be achieved by using compression techniques rather than simple decimation. However, latency and phasing of the digital video relative to voice and other data remain a concern.

Our PIs found they needed to experiment with different FRG only in the early sessions; thereafter they used combinations used in previous sessions.

Dynamic adjustment of FRG parameters imposed an extra workload on the PI, and PIs were reluctant to spend time re-optimizing once they felt they knew the appropriate FRG parameter choice.

We concluded that operator controlled FRG is best suited to situations where the observer needs high resolution both spatially and temporally. An example of such a task is remote manipulation. Sheridan and coworkers at MIT have previously shown the utility of observer controlled FRG in remote manipulation tasks. However, in our Telescience experiment, the video was used primarily for surveillance (during data-taking session of the OT experiment, the P.I. was given full video) so the P.I.'s never used dynamic FRG to its fullest advantage. Implementation of operator controlled FRG as part of the SSIS - should it be required for space station teleoperation - has the technical disadvantage that FRG commands must be up-linked from the PI to the space station video control.

Although our simulations demonstrated that PIs could accomplish their surveillance functions acceptably using slow scan video at 50 kB/sec and with frame rates as low as 0.25 Hz, we believe it would be a mistake for the SSIS to be designed so that only slow scan video is available to PIs. We believe that there are real advantages to having bursts of higher bandwidth video occasionally available to the PI to permit higher spatio-temporal monitoring of dynamic events. This was exemplified when we retrospectively discovered that our sled OKN moving stripe display unit had been consistently operating at the wrong speed through all the sessions in one of the experiments. The PIs felt that if they had been given occasional bursts of high bit rate video on demand, the display problem likely would have been discovered and corrected. We believe that PIs can accomplish most surveillance monitoring functions using black/white, relatively slow scan video. Further research is needed to establish the lower limits of acceptable frame rate, and appropriate methods for dynamically allocating available video bit rate between PIs.

- The integrated data and video (UNIX and Parallax Graphics board/X windows) environment we employed allowed us to rapidly prototype an "all glass" integrated, windowed, scrolling PI display using two coordinated screens which in several respects was a major advance over the displays our team had used in previous Spacelab missions (based on ANSI terminal, Tek 4014 display and paper stripchart technology). Our PIs could alter the size, number, and content of display windows to suit individual team preferences and the nature of the

experiment data, and rapidly scroll to and rescale data of interest. Most display control was accomplished via graphical buttons with appropriate functional icons and in some cases companion dialog boxes. The generic scrolling analog data window displays generally supported both experiments well, although PIs had numerous specific suggestions for improvements to the generic graphical interface, and many ideas they wanted to try to better tailor the software to the individual experiments. We believe it is important to develop software tools to support more rapid and inexpensive development of PI displays, not just for TTPP activities, but also for actual space station experiment development. It has been our experience in the Spacelab program that as PIs gain experience in early simulations, their technical requirements for data monitoring displays matures and changes. A set of graphical interface standards and a higher level software toolkit for development of PI displays to support rapid and reliable reprogramming of PI displays is needed which will take full advantage of the accumulated experience with the human factors aspects of interactive displays. We are concerned that although UNIX and X Windows provide powerful tools for software development, many PI team programmers may not have sufficient experience to put them quickly to effective use, and that UNIX/C/X Windows alone may be too low level a platform. A layered, high level tool approach to PI display software may have costs in terms of reduced data throughput and display performance, but the advantages in terms of flexibility and human factors operability are probably more important in most situations.

The ability of PIs to simultaneously monitor experiment operations, evaluate incoming data, and maintain an adequate running logbook was noted to be critically dependent on the division of labor and experience of the PIs involved, and on the software technology which supports them. Logkeeping and ground voice loop communications tasks often distracted our PIs. Also, it should be noted that increased PI workstation capabilities cannot be obtained entirely without cost in terms of increased PI team training time. User friendliness of the text editor used for logkeeping is also important. Further research is needed to develop improved techniques for computer assisted log-keeping, experiment progress status display, methods for comparison of current and "nominal" expected data, and hardware fault diagnosis.

2.10 Remote Operation of Spaceborne Microgravity Materials Science Experiments (RPI)

Experiment Description and Summary Conclusions

To determine, experimentally, the applicability of the Telescience (remote operation) concept to the area of Microgravity Materials Science (RPI), in cooperation with the Lewis Research Center (LeRC) Microgravity Materials Science Laboratory (MMSL), planned to establish a testbed to remotely operate experiments in the areas of; A) Glass Science, B) Crystal Growth from the Vapor, and C) Crystal Growth from the Liquid. Drs. Robert Doremus, Heribert Wiedemeier, and Martin Glicksman, respectively, having previous

experience in these fields, assisted in the program by advising on the type of experiments to be run and on the data and control requirements to make the experiments viable. Existing equipment at MMSL was to be utilized after suitable modification, communication links were to be established, and a remote control center was to be created at RPI. The program was planned in three phases (one year each). The first phase was to establish the required facilities, the second to operate the experiments to determine the limitations of the facilities and the third to upgrade the facilities and determine the facility requirements to allow the envisioned science return enhancement made possible by remote operation. These requirements would then be made available to Code S (Space Station) for possible use in the vehicle design.

Issue Investigated

The determination of the level of facilities required to provide a significant science return enhancement by remote experiment control. And, if possible, to determine a relationship between facilities level and science enhancement to allow a cost-effective determination of future equipment/facilities.

Experiment Hypothesis

In the opinion of the scientists participating in this program, based on previous experience, the scientific return of spaceborne experiments could be significantly enhanced if the Principle Investigator (PI) could have a "hands-on" relationship with his equipment while it is operating. In this mode he could make real-time adjustments to the experiment based on real-time results. This program is intended to determine the feasibility of this approach and the complexity of the required facilities.

Method of Investigation as Planned

To establish a test-bed to perform this experiment required three facilities:

1. Space type experiment hardware
2. Suitable communication links
3. A simulated remote control site

Experiment Hardware

The MMSL at LeRC was established to provide a facility containing a collection of spaceborne type materials science equipment which could be made available to potential P.I.'s to use to try new experiments to determine the feasibility of future flight hardware. In this manner new ideas could be justified for flight at a relatively low expenditure of time and money. These new ideas could then be considered for flight opportunities with a relatively high degree of success possibility.

In preliminary discussions with MMSL personnel it was determined that generic type equipment existed in all of the science areas contemplated by RPI. Some modifications would have to be made to this equipment to allow remote control and a communications facility would have to be established. Since the equipment existed, however, the cost of

establishing this facility could be kept to a minimum. MMSL agreed to cooperate with RPI and was separately funded to complete this effort as well as provide support during the actual experimental phase.

Communications Links

Three types of information flow are required for this experiment.

1. Data flow from the experiment to the control center
2. Command flow from the control center to the experiment
3. Video information from the experiment to the control center.

In the original concept for this program it was anticipated that the first two types could be handled by existing data networks (BITnet, ARPAnet, etc.). It was determined that the anticipated transmission delays would not be a problem. The video information flow was planned for the use of commercial communications satellites for high frame rates and landlines for slow scan transmissions. A major goal of this program was to determine the minimum amount of video data required in each of the three areas of science to be investigated.

Remote Control Center

RPI assumed the responsibility for establishing this facility. A small laboratory in the Materials Research Center at RPI was refurbished for this purpose. A Sun 3/60 computer was acquired for use as a control console and suitable video equipment procured for the display of the video information. It was anticipated that existing, user friendly, software programs (OASIS from the University of Colorado for example) could be modified for our use. In addition RIACS, as its contribution to the program, was to provide software support and additional programs as required. The requirements for software development at RPI would, therefore, be minimal. One of RPI's major responsibilities was to assure the compatibility of the MMSL software and the RPI software.

Method of Investigation as it Evolved

During the first four months of this program many major problems were encountered which required several changes in program direction and caused significant delays. Among these were:

- Delays in acquiring the Sun 3/60
- Problems with existing data networks which precluded their use for this program
- Incompatibility of existing software programs with the Sun UNIX operating system
- Lack of video transmission capability for commercial satellites at LeRC
- Equipment procurement delays at LeRC
- Requirements for software development at RPI beyond those anticipated and the development was delayed by the Sun delivery delays

Because of these problems a revised plan of action was prepared to maximize the first phase results. This plan included:

- The establishment, at RPI, of a complete end- to-end system. This required the acquisition of video equipment, computer software, and experimental equipment not originally anticipated. This change was made to facilitate the complete debugging of the system at RPI to minimize the use of expensive landline charges and reduce the cooperative support needed at LeRC.
- The procurement, at RPI, of a satellite receiving station to interface with the NASA teleconferencing system which was to be used for the video transmission.
- The development, at LeRC, of a computer controlled video frame acquisition and data compression system to allow frame transmission on deal-up telephone lines at 9600 baud rates. A study of data compression, at RPI, indicated that compression of both resolution and brightness was possible without serious loss of useful information.
- Reduction in the complexity of the experimental equipment at LeRC to be used in this first phase of the program.
- The establishment of a communications protocol and software to allow the interfacing of two operating systems; UNIX at RPI and MS-DOS at LeRC.

Experimental Results

To restate the goals of first phase:

- Establish a telescience (remote control) testbed specifically aimed at Materials Science requirements.
- Obtain preliminary results from the operation of typical materials science experiments.

As of this writing (September 1988) one month remains on the original contract and the status of the program can be stated as follows:

% Complete

- Establish a remote control center at RPI - 98%
- Establish an experiment center at LeRC - 60%
- Establish a communications protocol - 90%
- Establish a communication link
 - Command/Data - 90%
 - Video - 80%

- Establish a remote control - 80% experiment @ RPI
- Operate a full-up system - 50%
- Perform analytical studies - 90 %

The following results have been obtained to-date:

- Existing communications capabilities, both land based and space, are not adequate for real time experiment control.
- New and more extensive data compression schemes are needed to support this concept.
- The development of new, higher resolution, video equipment is required for many materials science experiments.
- The concept of telescience for materials science should be separated into two specific areas. The first, remote control, requires moderate levels of communications but also has unique problems with controls on experiment operation to assure spacecraft safety, crew safety, experiment interaction, and resource allocation. The second, data acquisition and transfer, requires very high levels of communication capabilities and data acquisition capabilities. Both are important to telescience but the concept does not require both to be viable.
- The concept of telescience has been accepted by many of the experimenters in the field to enhance the scientific return. Previous attempts to enhance scientific returns by using highly automated equipment and/or hands-on crew assistance has been moderately successful but experience has shown that the expertise of the investigator cannot be transferred to either a machine or crewman under the practical limitations imposed by the space program. Often, rudimentary data and command flow capabilities can make the difference between an experiment's success or failure.
- Establishment of adequate communications along with the appropriate equipment modifications are not incompatible with the Space Station guidelines.
- Because of the level of complexity involved it is obvious that there are some planned materials science experiments that are not compatible with the telescience concept. The majority, however, could have their science return increased significantly and their chance of success enhanced by the use of telescience concepts.
- Video requirements can be reduced to less than a 256 x 256 pixel resolution and less than 16 shades of gray and less than the standard 30 frame per second rate for many remote control operations without degradation of usefulness.

2.11 Remote Operation of A Science Instrument (Smithsonian Astrophysical Observatory)

Issue to Investigate

Our testbed activity was to show how one would remotely interact with an array over a network connection and make modifications to the software, control the operations, remotely view the data acquisition in realtime and finally transfer the data set to the analysis facility over the network. This was then to be compared to the prior approach of performing this task.

Previously, debugging and program modifications would either require the persons to be located with the instrument and computers (incurring substantial travel expenses) or otherwise result in lengthy delays due to getting information and problem reports to the relevant engineers and programmers. The latter approach was a laborious way to do the debugging with the programmer on one end of a phone line and the scientist at the other. Completing a network link and thereby enabling telescience to the mountain top would result in more efficient use of human and system resources. Further system improvements and upgrades could also be carried out more efficiently. Thus, without even having used the link for teleanalysis, the improved operations and software development will have increased the functionality and productivity of the scientific, engineering and programming staff, an overall objective of telescience. This end-to-end system of instrument to scientist will be very analogous to what is envisioned for the eventual space flight configuration.

Experiment Background

As a result of the detector development work being performed by SAO for the SIRTF program, IR arrays are becoming available for ground based evaluation and use. These arrays will be analogous to many of the types of arrays that others may use for various experiments to be performed on the Space Station. Operationally, the arrays will be controlled locally with a microprocessor and the two-dimensional data will be sent via a serial stream to the ground to be processed. The investigator will then analyze the data and, based upon the results, modify the next portion of the observing program or the instrument control parameters.

SAO purchased with Institution funds, an InSb 1-5 micron 58x62 pixel infrared array for use at its ground based telescopes. The control and readout electronics, dewar for cooling the array, mechanical refrigeration system and computer system for operation of the array and recording of the data have been developed independent of the telescience testbed activities. However, the Telescience Testbed activity provided the opportunity to place the processor used for operation of the array on-line via networks, thereby, connecting the observing site with the remotely located scientists and support staff.

To accomplish this we completed the network connection between the instrument computer at the telescope and the data analysis facilities in Cambridge, MA with the construction of a T1 link between the telescope and the University of Arizona campus. The Cambridge facilities were already linked to the SAO microVAX at Tucson, AZ using the Internet. During the past year this link was to be enhanced with 56k baud links for the JVNC

connection since both Harvard and University of Arizona are part of the NSF supercomputer project and via 56k link as part of the NSN. (However, neither of the links have been completed as of the end of these telescience activities.) SAO owns a microwave link between the MMT on the summit of Mt. Hopkins and Steward Observatory in Tucson.

The plan was then with the link in place, the IR array could be operated much as an array in the Space Station era would be operated. During the commissioning period of the array the engineers and programmers would remain in Cambridge, much as if in a space flight operations center, using the network to perform porting of programs to the telescope and debugging by remotely logging onto the microVAX at the mountain. Once debugging was completed, the operation would then take on a semblance of a typical space flight operation, with an instrument operator at the telescope (for attached payloads in space, an astronaut would perform this function), the scientist remaining at his or her home institution, and the images being transferred via network to the image processing facility at the scientist's home institution: the true embodiment of telescience.

This testbed would then have wrung out many of the issues that need to be addressed with regard to teleoperation and teleanalysis.

Experiment Testbed Results

There were actually five separate processes that had to be accomplished before the results for this Telescience activity could be achieved:

1. The T1 link to the mountain had to be built;
2. The instrument control software had to be written;
3. The near infrared array camera had to be built;
4. All the above had to be brought together and made to work at the telescope before any images could be viewed from Cambridge.
5. Finally, data images could be transferred from the telescope to Cambridge for analysis.

The first activity of establishing the T1 link between the mountain and the network gateway on the University of Arizona campus was rather straight-forward but also involved considerable effort to accomplish. We made use of existing spares for the microwave link to provide for a new bi-directional link between Tucson and the mountain. To complete the network link required procurement and installation of T1 adapters for the microwave system and T1 interfaces for the microVAXes. The work on the microwave link was completed by January with the T1 connection from Tucson to the mountain and back working in the loopback mode. The microVAXes on the mountain and in Tucson were then connected via T1 and via wide area network to Cambridge. A class B node assignment was obtained for the microVAX on the University of Arizona campus network and class C nodes for our local area network (LAN) were established in Tucson and on the mountain. Figure 1 illustrates the link between the Tucson LAN and the LAN on the mountain. Additional work was needed to make the DSUs and ACCs in the VAX on the mountain work. We were finally able to remotely login on the VAX on the mountain from Cambridge via the Internet by mid-April.

The next major element of the system was the realtime instrument control software for operation of the array. This software was designed to run on a VAXstation II/GPX under Ultrix. The system allows the combination in a relatively compact package of several text displays, a Tektronix vector graphics terminal and an imaging terminal. This is all made possible by the mouse, keyboard and bit-mapped display (1024x864 pixels by 8 bits). The various windows perform the emulations of different kinds of terminals and handle the interactions with the user by the X Window system. This system was developed to support the IR array as well as for the future Reticons and CCD cameras at SAO.

The IR array data collection routines are a suite of separate programs, running as simultaneous processes, that pass messages back and forth via pipes, message queues and shared memory, as appropriate, to control the instrument. The command interpreter, vector graphics, image display, and status display programs use the network-transparent X Window System (tm) for output. In addition, the status and image display programs take as input a shared memory block, i.e., a block of memory that can be accessed simultaneously by other processes in the system, especially the data-taking ones.

In mid-September this software, amounting to about 2.5 Mbytes of files, was transferred over the Internet to the VAX on the mountain. The transfer took several hours with an average throughput on the net of about 1.4 kbytes/sec in the middle of the day. The connection was lost twice during this time period. The software was then installed on the host at the mountain and the kernel rebuilt. Unfortunately, the communications board was not configured correctly and the link lost in the process. An on-site technician quickly fixed the problem. This completed the second major element for the testbed activity.

The IR array development was the final pacing element. SAO purchase with institution funds the IR InSb array for this ground based camera. SAO then built the control and readout electronics, dewar for cooling the array, mechanical refrigeration system as well as the realtime instrument control software described above. Most of the hardware was pulled together in Spring and it was anticipated that the first run of the new array at the telescope could take place in June. However, the complete system was not working satisfactorily in June. The first use at the telescope took place in September.

The actual telescience results came from the experiences of the first use of the detector system, referred to as SONIC (Smithsonian Observatory near Infrared Camera), at the telescope.

As it happened, the person who had developed the software system did not make the trip to Tucson (he was getting married that weekend) and therefore remained in Cambridge. Since he could not go the mountain, the run would have failed completely were it not for the network link. An "unplanned experiment" without the use of the network was the prime example of performing "teledebugging". When the technician called up with an instrument problem, the link was (temporarily) unavailable and the software programmer had to try to talk him through some tests over the phone. Essentially nothing was accomplished over the telephone. When the link came back and he could login over the network, the programmer could investigate and solve the problem in minutes (or perhaps an hour), even given the limited bandwidth and latency. Without the network, he never could have otherwise solved the problems except by physically being at the telescope. This experience showed that it takes almost no time to become thoroughly dependent upon network access.

The software architecture described above allows one to start up a IR array data-taking session on the VAXstation at the telescope that directly displays on the VAXstation in Cambridge exactly as it would have on the mountain. These displays include a command input window, a background message window (mostly for debugging messages), an instrument status display, updated once per second, and finally a quick-look image display window showing the IR image as it integrates in computer memory. The data recording are still done at the mountain, but the viewing can be done remotely over the network.

Of course, there are limitations to this setup: the network bandwidth and latency across the >2,800 or so miles (and unknown numbers of gateways) of the Internet limit the speed at which the integrating image could be updated, and the responsiveness to commands typed in Cambridge. In addition, the network would spontaneously disconnect about once each hour or two. Nonetheless, one was able to make use of this arrangement in Cambridge to check out (and correct) several minor changes to the software, immediately testing their effects on the hardware at the telescope. It was really pretty thrilling to watch an image integrating in realtime on an instrument located on a mountain 2,800 miles away.

Another useful mode of operation was to 'spy' on the data-taking computer running the system. Since the IR image and status information are stored in shared memory blocks, one could simply log in and execute independent status and image display processes, outputting them to the system in Cambridge. One could thus keep track of the instrument and current integration while examining and altering ones own copy of the quick-look IR image independent of the instrument control computer. One could have also started a command interpreter that would have sent commands to the instrument control processes asynchronous to the instrument control computer.

Finally, we demonstrated transfer of data from the mountain to Cambridge after the run. We did not make use of real-time transfer because there was no one in Cambridge during this first test of the system to reduce the data. For the future, we will not have to duplicate the data reduction programs, which take quite a lot of storage space, rather, we will transfer the data to Cambridge as we did for this first run, reduce them, and transfer the results back to the astronomer at Mt. Hopkins. The reduction can be done either by someone in Cambridge or by the astronomer at Mt. Hopkins via remote login.

Experiment Conclusions

All in all the experiment turned out not only to be successful, but in addition we have shown that one can conveniently port software, rebuild the kernel remotely (if done carefully), debug any problems remotely and in realtime and watch the data from the image accumulate in realtime. Finally, the entire data set can be sent back to Cambridge for analysis over the network link.

2.12 Remote Commanding and Telemetry Reception (UC Berkeley)

One of the important goals of the telescience testbed program is to develop methodologies to allow scientists to interact with space-based instruments as they would in a laboratory environment. To this end, we have undertaken a teleoperation project to determine if currently available technologies are mature enough to support real-time operation with instruments from a remote site.

Our teleoperation experiments used the EUVE payload which is under development for a 1991 launch. They demonstrate transparent command and telemetry handling over various computer networks. Details of the experiments and results can be found in [Chakrabarti et al., 1988b].

Experiment Description and Summary

A hardware/software simulator for the EUVE, called Kiwi (for the bird that does not fly), was developed for EUVE. The Kiwi simulates the science payload as well as the spacecraft and was used in our experiments. The spacecraft was simulated on an IBM PC equipped with special hardware and software. A simulation of a Science Operations Center (SOC) was created on a Sun Microsystems workstation which was connected on the SSL LAN. In this way, the instrument on a spacecraft is treated like a node in a computer network.

The Kiwi was always located at the SSL for all experiments. The user interacted with the kiwi from:

- another workstation on the SSL LAN,
- a microVAX workstation running the Ultrix operating system located at the Stanford University, and
- a Sun Microsystems workstation located at the University of Colorado at Boulder.

The user sent commands to the EUVE Kiwi from the workstations described above, and received telemetry from it. In the last experiment a video camera was used at the Kiwi to provide a visual confirmation of the execution of commands to Kiwi.

Issues Investigated

The first issue from these experiments indicate that current technology is adequate to support transparent teleoperations. We have run the experiment using workstations from different manufacturers. Our experience indicates that the use of the UNIX operating system provides some degree of vendor independence. However, various versions of UNIX are different enough that it was not possible to run the experiments without modifications to the software or the operating environment.

These experiments have highlighted a number of issues related to the applications of computer networks in space-based research. Continuous access to network resources is mandatory for realistic teleoperations. Our experiments using the Internet were severely hampered by the poor reliability and low effective bandwidth of the channels. In running our demonstration between Boulder and Berkeley, for example, it often took several attempts to establish the connection to UCB. Once connected, we would experience generally good throughput for 5 - 20 minutes or so, but then the connection would be broken. A conclusion may be drawn that existing networks are adequate for electronic mail, block file transfer, and terminals but are inadequate to support the continuous bidirectional data flow necessary to support telescience. Current network technology is adequate if specific links are dedicated to operation of instruments.

Software, and to a lesser degree, hardware compatibility still remain major problems. We attempted to run the teleoperation experiment from a DEC VAX 11/780 computer running the Eunice operating system. Eunice provides one version of the UNIX operating system environment which runs on top of the VMS operating system. Although Sun's UNIX and Eunice appear to users to be very similar, we were unable to run the testbed with the clients running Eunice. We investigated the problem and determined that it was due to the security system's refusal to allow us to change the UNIX socket configuration. With proper access, one should be able to overcome this hurdle. However, we have successfully conducted the experiment with the client environment running Ultrix on a microVAX.

Using the UNIX operating system for both clients and servers has helped ameliorate some of these problems but an enhanced effort is required to develop networking tools that are independent of language and operating system. These include programming language bindings and basic packaged tools.

Command encryption and validation will be required during flight operations. We have made a small attempt to address the data link security control needs. For the time being we are using the UNIX "crypt" program for testing purposes. Although "crypt" is not as secure as the Data Encryption Standard (DES), it provides a test of the software overhead involved in command encryption. This overhead can be substantial: during the Boulder demonstration it was necessary to disable the UNIX encryption tool crypt in order to make sure all the commands were correctly received and executed. Software implementations of DES, for instance, did not appear to have the speed to support telescience data rates. Sun Microsystems has provisions for hardware support of DES, but we need to evaluate capabilities offered by equipment from other vendors. We believe that further work needs to be done to standardize the encryption protocols to be used over telescience data links. We also note that the U.S. Government restricts the distribution of DES hardware and software outside the United States.

Experiment Hypothesis

The hypotheses for the teleoperation experiments were:

- Technology is mature enough to support routine interactions with instruments in a laboratory-like manner.
- The computer networks are reliable enough to run the tests. Furthermore, the network bandwidth is sufficient to run the tests, provided data compression is applied to the telemetered data and video images. We also assumed that the delay of the internet is representative of those encountered during interactions with a space-based instrument.

Method of Investigation

Our teleoperation testbed concentrates on the operation of the EUVE instruments over the NSI network. The Kiwi can be reached through a Sun Microsystems workstation that is connected to the SSL's LAN so instrument commands can be sent to the CDP from any workstation connected to this network. This is not done with a simple remote login: servers on each workstation exchange data packets directly using TCP/IP.

The Kiwi is connected to the outside world through command and telemetry server workstation. A single-board 68000 computer acts as the hardware interface to the Kiwi. It buffers all telemetry and command requests. The server machine contains the telemetry and command servers, which are awakened by appropriate service requests from the clients (users). The clients can be conceived as operations requested from a SOC. The system responds in the same manner for a request from a machine on the SSL LAN as for a request from a machine anywhere on the Internet; all that is required is that the machines and users have appropriate access permissions.

Each telemetry request is assigned a telemetry slave, but only one command slave exists at a given time. Subsequent command requests are queued. When control of the instrument is relinquished by the current commander, the next command request is served.

The purpose of the client-slave process pair is to mask, from the user, the physical implementation of accessing the payload. A client allows the user to treat a payload as a local, dedicated resource. This frees the user and tool developer from the specific development state of a payload. When tools are independent of the communications medium, they will function during development (when the breadboard payload is on a bench next to the user), or when the payload is on orbit.

A client provides only basic access to a payload at the lowest layer. Tools must be built to control the masses of data flowing to and from a client. Some of those can be utilized by many users; some of them will be unique to an individual user.

The UC Berkeley campus is connected to several of the TCP/IP networks that make up the Internet: NSFnet (National Science Foundation Network), ARPAnet (Advanced Research Projects Agency Network), and BARRnet (Bay Area Regional Research Network). Gateways to non-TCP/IP networks such as Bitnet, SPAN (Space Physics Analysis Network), and Usenet are also available. The SSL LAN is connected to the campus network through a 56K bps link.

After performing the remote operation experiment from a foreign host on our LAN, we conducted the same experiment over a wide area network. We installed the telemetry and command clients on a Digital Equipment Corporation microVAX computer running the Ultrix operating system. We then ran an experiment in which commands and telemetry were exchanged between the Stanford computer and the Kiwi at UCB. All of the commands were received and executed properly. This experiment demonstrated that, in addition to interacting with an instrument from a remote site, we were able to achieve some vendor-independence.

In a second remote operation experiment, we operated the instrument from a Sun workstation located at the University of Colorado in Boulder. The experiment was demonstrated at the TTPP2 meeting held at the University of Colorado, Boulder. Command and telemetry clients were run on the Boulder machine. The instrument remained on the UCB LAN and was connected to the client machines through the Internet. We established a command stream to operate the instrument and monitored the telemetry to verify proper execution. All commands received were executed properly.

The telemetry rate of the EUVE experiment is 32 kilobits per second, but data compression reduced this considerably for the demonstration: only engineering data were sent, so the telemetry stream was relatively empty. It is difficult to gauge the actual network throughput attained, but we estimate that we averaged approximately 4 kilobits per second, with occasional bursts of up to 8 kilobits per second between UCB and Boulder.

Data compression was achieved with the UNIX compress command. It uses an adaptive Lempel-Ziv compression scheme that provides good error-free compression with reasonable computational overhead [Welch, 1984]. At the client end we used the UNIX uncompress command to regain the original data. Standard UNIX tools allowed us to achieve useful data compression, although they may not have provided the highest possible compression. The same data compression scheme was used for both the telemetered digital data from the instrument as well as the accompanying video signals (described later).

The primary purpose of this experiment was to operate a d.c. motor that is connected to a door covering a high-vacuum cavity. During flight, two micro-switch monitors indicate the position of the motor; for the teleoperation demonstration, we decided it would be useful to use a video monitor to confirm the effects of the command execution. The use of video signals over the Internet concurrently with the command and telemetry signal was our first step to test the present network infrastructure. We expect that similar video images will play an important role in space experiments in the space station era.

Time and funding constraints dictated that inexpensive, off-the-shelf tools be used for video support. We selected the Imagewise video digitizing system made by Micromint, Inc., of Vernon, Connecticut. It consists of a digitizer/transmitter that collects individual video frames and transmits them over an RS-232 line to a receiver/display unit that converts the data back to a standard NTSC video signal for display on a monitor. This system provided us with a low-frame-rate video image, which was then sent over the Internet for visual confirmation of the telemetered door position values. The digitizing system can be operated at several different spatial resolutions and gray levels. The system uses a run-length encoding scheme for data compression. Operating at 19.2K bps with a resolution of 128 by 122 pixels, it can transmit a video frame in 10 seconds or less. This was sufficient for the experiment - viewers could see images of the mechanism before and after the commands were sent.

Some additional software was written to allow us to transmit the video images over the Internet. The Imagewise transmitter sent video frames to one of the workstations in the SSL LAN, where a server program performed some additional data compression and transmitted the compressed data to a client at the remote site in Boulder. The Imagewise receiver and monitor were not used in this experiment. Instead, the client software uncompressed the bit stream and displayed each video frame in a window on the console screen of the Sun color workstation. A display program called "imtool," developed at the National Optical Astronomy Observatories for their IRAF image analysis system, provided a ready-made display window for the video images [Tody, 1986].

Experiment Results

All commands and telemetry were properly received in all three experiments described above. We have, however, discovered some problems in running the tests as described in the Issues Investigated section. The video images were also received over the internet during peak use hours as was demonstrated at the Boulder meeting. Although, the network reliability and delay caused some irritation, as described in the Issues Investigated section.

Section 3

Teleanalysis Experiments

3.1 Telescience Field Campaign (Univ. of Colorado)

The objective of this subtask was to demonstrate the use of selected telescience techniques and capabilities to improve the conduct of an earth system science investigation in a field environment.

Experiment Description and Summary Conclusions

One objective of this University of Colorado research is to improve our estimate of the level of solar activity as Solar Cycle 22 builds. Activity in Cycle 22 will determine the space environment for new satellites to be launched within the next five years, i.e., UARS and the Hubble Space Telescope, as well as presently existing and productive ones such as SMM and SME. An estimate of the strength of the next solar maximum in 1990-1992 is important in predicting the satellite orbital lifetimes.

This particular task centers around a SME Line Analysis Program being conducted by Dr. Oran R. White from his ranch in southwestern Colorado. He analyzes the time series of solar line intensities measured from spectra of the full solar disk by the LASP UV spectrometer on SME. The SME database also contains time series of other solar activity measures such as sunspot number and the 10.7 cm radio flux, which are studied together with the SME measurements. He uses a personal computer both for local processing and as a "smart" terminal linked to a larger LASP host computer.

In summary, it is concluded that this mode of operation made a major improvement in Dr. White's productivity. In fact, it is uniquely this capability which makes it possible for this valued researcher to continue his contributions to the LASP research program. By simple extension, this mode of operation is appropriate, both for researchers well removed from sites having mid- or large-scale computers, data repositories, and working databases, as well as for researchers at the larger centers who may find this approach more productive than working with the more traditional "dumb" terminals.

Issue Investigated

What is the effectiveness of a very simple, inexpensive workstation centered around a standard IBM PC XT computer in working remotely with the SME database in conducting a scientific investigation?

Experiment Hypothesis

It should be possible, using very inexpensive, commercially available computer equipment and software, to make a substantial contribution to the LASP/NASA research program from a distance.

Method of Investigation

Dr. White, a SME Guest Investigator, collaborates with Dr. Gary Rottman at LASP and conducts his research from a ranch in southwestern Colorado. He uses an IBM PC XT computer connected from his business to the LASP VAX computer via commercial telephone lines. He accesses the SME database through codes written in the IDL programming language, performs data manipulations and computing in the VAX computer at LASP, transfers the intermediate graphical and tabular results to his PC, and does the final preparation of the data and analysis at his home site.

Experiment Result

The effort was fully successful. A LASP research report entitled "SME as a Testbed for Telescience - A Case Study" is being produced as a TTPP product. This report includes a more detailed discussion of the methodology, equipment and software, and scientific and technical results

Some of the more important lessons learned from this task include:

1. In a programmatic sense, future testbed plans must include adequate provisions (including identifiable funding sources) for both the technological and scientific content of the work.
2. Data and information retrieval from the host computer by the remote site for analysis purposes is workable over commercial telephone lines using either a file transfer system in the remote terminal emulator or by direct screen display and logging.
3. On-Line analysis works very well if the user is properly prepared and parcels the work into time segments of less than 45 minutes per session. Longer sessions result in fatigue and an increased error rate.
4. A key component of the remote system is a good terminal emulator (both text and graphics) to permit the remote system to connect to the VAX host as a standard terminal. Graphical display is crucial to productive user interaction.
5. An assortment of highly effective processing tools (e.g., IDL) on the host computer are necessary to minimize telephone line costs. Quick access to simple utilities such as editors, loggers, and communications software on both the remote and host computers is also crucial.
6. Improved data manipulation, presentation, and analysis tools for the remote PC terminals are needed, and substantial developmental efforts are warranted to produce them. The emphasis here should be on efficient personal interaction rather than remote-site large-scale batch processing.

7. This task reached the practical limits of the PC XT. The immediate limiting factor was the 640 KB memory, but the speed factor was close behind. It is clear that a remote computer with a memory capacity exceeding 1 MB and a 16 MHz cycle time is necessary for a true stand-alone machine capable of staying in step with, for example, the IDL processing.
8. The use of commercial voice quality telephone lines does present some noise problems for the on-line user. This usually shows up immediately upon logon, when the standard reaction was to hang up and redial. Once a clean connection was established the error rate tended to remain good. Although noise does not pose fundamental problems in the analysis, it does lead to some inconvenience, frustration, and added expense
9. A good electronic mail facility is an absolutely essential adjunct for telescience.
10. The availability of archival data on high density media such as optical disks may lessen the dependence on the telephone line, and should open an interesting new domain for telescience.

3.2 Remote Access to the SME Database (Univ. of Colorado)

The objective of this task was to improve the capability for remote access to the LASP SME database to make it easier for the consortium members and other outside-LASP researchers to use these data in collaborative research while working from their home sites.

Experiment Description and Summary Conclusion

LASP developed a data processing system as a part of its original SME project activities. This included an extensive set of databases, along with software to access and manipulate them. One of these databases, SMEDATA, is accessible via the SME MENU program. Although this system has been useful for the internal LASP researchers, it had limitations which made it difficult for outside researchers to use it. In this telescience task the SME database was improved, and the SME MENU system was enhanced to provide improved accessibility.

This work has been completed. A User's Guide is being produced as one of the TTPP products. This User's Guide will make it easy for any telescience collaborator with access to a modem and telephone line to make inquiries, browse, manipulate, and extract data and derived information from the SME database.

Issue Investigated

How can the SME database and access system be improved to make it possible for an investigator anywhere in the country to easily gain access to and use SME data for collaborative research?

Experiment Hypothesis

It should be possible and relatively straightforward for researchers outside the SME project to use the SME data.

Method of Investigation

The data set structure was modified by Barry Knapp to make the database more usable by a wide variety of users. Julie Dawe completed a set of changes to the SME MENU program to handle this modification of the data set structure. And Barry Knapp completed the cleaning up and integrating of the Interactive Data Language (IDL) routines which are called by the SME MENU program. The specific changes included the following:

- Addition of complete on-line descriptions of each data product (physical units, record structures, ranges and resolutions).
- Automatic determination and presentation of ranges of data availability.
- Improved "screen economy" to maximize the amount of data displayed per screen, hence per transmitted line during remote screen capture.
- Automatic assembly of user-selected data subsets into compact, formatted, downloadable files.
- Development and installation of a library of self-documenting IDL routines for access to and analysis of the SME data products, for those users who can use IDL.
- Easier, more logical menu navigation.
- Improved robustness and error-recovery.

The SME Science Data User's Guide has been greatly expanded and rewritten by Pieter Kallemeijn, Barry Knapp, and George Ludwig to make it possible for scientists not familiar with the SME data system to acquire and use these data in their investigations.

Experiment Results

As a result of these actions, the MENU system is considerably easier to use, smoother, and more dependable in its operation. All of the SME final science data products are now up to date and accessible on line within a week or two of data acquisition. This provides a far more suitable environment for distributed users in a telescience environment.

3.3 Remote Data Analysis (Cornell University)

Investigation of the problems of transporting software from an already established data analysis system to a "more modern" one show that this may not be a cost effective solution even though the initial manpower investment may have been quite large. The establishment of standards early-on in the development phase is necessary for a smooth transition between institutions. The problem of establishing network connections and building-up a level expertise on new system is difficult in small research groups. Indeed, the objective to perform remote data analysis and software development with the University of Rochester was hampered by problems establishing network connections to local computers. The SIRTF project has benefited via the telescience effort through the establishment of common

hardware/software guidelines and the experience gained in dealing with networked systems. Knowledge of the strengths and limitations of networks will be extremely useful in designing the SIRTF Operations Center.

Remote Data Analysis

Cornell has worked in conjunction with the University of Rochester to perform remote software development and data analysis of infrared image array data.

Experiment Description and Summary

The objective of the Cornell telescience effort was to perform remote access, reduction and analysis on data obtained with the University of Rochester infrared array camera. The data analysis software and data was located at Rochester. The initial plan was to perform this task remotely. The challenge of this effort is in the diverse computer resources and programs involved and in the fact that the techniques and ideas for data analysis must be transferred remotely.

The experiment was hampered by the difficulties in linking into campus networks. In fact, this linkage was delayed indefinitely at Rochester due to problems in running cables through tunnels containing asbestos. The result is that our initial experimental objectives had to be modified. Our problem became one of how best to effectively port the analysis software available at Rochester to Cornell. This is a non-trivial task because the Rochester software runs on a PDP 11 under the FORTH programming language.

To summarize the results, the problems of porting software have been most difficult and are those that indeed seem obvious. Lack of planning of portability in the early stages can result in great difficulties in the future. This has tremendous application to NASA's large scale projects which involve different instrument teams delivering software to the Operations Center for running instruments and analyzing data. With the impetus provided by our experience in the TTPP, the SIRTF instrument teams have achieved the first step of commonality by defining hardware and software standards for instrument control and data analysis. This will make the transition of the knowledge base to the Operations Center much smoother and less costly.

Issue Investigated

What is the feasibility of performing data analysis and developing software remotely with an already well established system (but one which was not designed originally for this task).

Experiment Hypothesis

It is surmised that porting a mature analysis system to a new environment (a Sun in this case) could be more efficient than transferring the techniques (algorithms) to run on the Sun under an established front end such as IRAF (Image Reduction and Analysis Facility).

Method of Investigation

Based on our experience of trying to run IRAF remotely over the network with the University of Arizona (which demonstrated this is not feasible), we decided that it was better to have analysis software at the home institution where the analysis is being done. The data can be transferred over the network (assuming it is not too large a data set) and software development can be reciprocal in this fashion also.

Work by a programmer employed part-time has centered on developing software to enable us to handle the Rochester data. What became evident in this process is the need for standardization. Our experience emphatically demonstrates this. The University of Rochester experiment runs on a PDP 11/15 computer. This selection is a bit historical in nature, since at the time the selection was made limited computers were available that had sufficient processing speed and commercially available fast A/D boards to be able to handle the data rates at reasonable cost. The problem of handing real time operations was also an issue. As in many "early" astronomical applications FORTH is used for both data acquisition and data processing.

The tasks then were several fold, transferring data between the Rochester PDP 11 computers and the Sun 3/60 at Rochester (purchased via the telescience program), transferring data to Cornell, and analyzing the data here on a Sun 3/60. This has to be done while maintaining data integrity and performing the proper operations for data analysis. It should be pointed out that the data analysis procedure is nontrivial, involving many operations. Our selection for the simplest means to transfer data between the PDP 11 and Sun computer is to use Kermit. Because of problems with linking the Rochester Sun to their campus network pointed out earlier, we could not link directly between Cornell and Rochester, and various means are being used to communicate data and code back and forth. This includes hand carrying tapes.

To perform the analysis at Cornell we were then faced with the task of either rewriting the analysis software in a new high level language (e.g. C or Fortran), writing analysis routine to work under some already established image processing environment such as IRAF, or bringing up FORTH on the Sun and port Rochester's current analysis software from the PDP 11 to the Sun. The first of these tasks was ruled out immediately because of the complexity of the task and the manpower required. Rochester has invested many man years of effort in the development of their software. As it turns out IRAF is not well suited to handling many small array images (32x32 for the Rochester array) and does not have many of the routines necessary for the reduction process. Upon discussions with our programmer it was decided to bring up FORTH on the Sun and try to port Rochester's software to the Sun. Unfortunately this effort has achieved only marginal success because of the difficulty of establishing a graphics interface with FORTH.

Our selection of choosing FORTH as a programming language does not seem wise from an aesthetic viewpoint however from an implementation aspect it logically appeared to be the quickest route and the code would then be portable to other Suns (and hopefully to other UNIX based systems). After processing with the FORTH code, output is in a standard format, e.g. FITS, so that IRAF can be used for some final analysis, display and plotting.

Experiment Summary

In summary, we concluded that having common hardware would have alleviated the problem of porting software but this is not a practical solution since it would not take advantage of current technology. The real solution nowadays is the selection of software and/or hardware which allows easy upward migration in the future as technology evolves. This is especially true with projects such as SIRTF which will take 10-15 years from team selections until flight.

As a footnote, the difficulties of linking into networks and getting up to speed on such systems were much more severe than anticipated especially in our small research group in which manpower is limited. The added bonus however is that the experience gained is extremely beneficial in understanding how networks and remote operations will have an impact on the design of the science operations center for SIRTF.

3.4 Telescience Occultation Observation (MIT)

Telescience concepts, methods and hardware were incorporated into a scientific observing program. MIT astronomy has conducted an experiment to use telescience methodologies in a very intense and successful occultation observation.

Experiment Description and Summary

We investigated the benefits and or drawbacks to working in a real time environment where the addition of telescience methodologies could be evaluated as to their usefulness in terms of the success or failure of the experiment itself. This we believe to be an acid test of the applicability of the methodologies.

We used the occultation of a star by the planet Pluto (a rare event) as the scientific experiment (vs the telescience experiment) that would set the stage for this telescience test. The science experiment was very complex and required significant feedback to the investigators on a daily and at some points an hourly and minute by minute basis. Communication with collaborators in Flagstaff Arizona was also required.

The science experiment may be divided into three different pieces for the purposes of this discussion: predictions, data taking, data reduction.

Predictions

This phase consisted of determining where on the earth to deploy the KAO flying observatory to observe the occultation. In order to make the predictions, astrometry of Pluto and the star to be occulted was performed.

Data Taking

The data was taken with an imaging ccd photometer (MIT SNAPSHOT camera) placed aboard the KAO flying observatory. This consisted of two flights the first being a calibration and check out flight.

Data Reduction/Analysis

The data reduction took place on the MIT campus primarily in a VAX -750 and from there ethernetted to a microVAX II GPX for archival onto an optical disc. Analysis started at this point between ourselves and collaborators.

Experiment Summary (Science)

Software Development, June 1987 - January 1988

From June 1987 to May 1988, a team of 10 people (3 MIT staff, 2 Lowell staff, 2 graduate students, and 3 undergraduates) worked part time to define the software, hardware, and data needs of the Pluto occultation project. The preliminary planning was done from June to September 1987 at both MIT and Lowell Observatory in Flagstaff Arizona. From September 1987 to January 1988, all phases of the project were planned in detail: necessary software was identified and specifications were laid out, changes to the CCD and data recording system were planned, and MKO was chosen as the observing site. During this phase of the planning, we had 7 people each working between 5 and 20 hours per week .

January - May 1988

From January to May 1988, the software plans were finalized, the programs were coded, and the data reduction procedures were decided. The data were in the form of spacewatch "strips": the telescope was locked into position, and the CCD was clocked out at the same rate at which the sky was passing by the chip. This results in a strip of data which is as wide as the CCD chip, and can be as long as the data recording system can handle. The data reduction "pipeline" was designed to handle these spacewatch data strips which would be arriving from MKO at the rate of about 15 strips per night, with 8 Mb per strip. These data would enter the pipeline as strips, and after being processed by 10 programs, would exit as "star lists," row and column coordinates of each previously chosen network star in the strip. The end of the pipeline involved checking the star lists for any obvious problems and then archiving the original data and the intermediate results onto OPTICAL DISKS The planning, coding, and implementation of this pipeline was done by 3 people working 20 to 40+ hours per week.

MKO Observing

Set-up: The CCD went out to MKO on 17 April 1988. The first data were taken on 23 April, and arrives at MIT on 28 April. Focus and coma problems are solved, and first astrometric data is taken on 1 May 1988.

Data obtained: Data were taken on 31 of the 41 nights from 23 April to 2 June 1988, averaging 5 magnetic tapes per night (3 strips per tape). 160 tapes (25 Mb per tape) were recorded, for a total of 4000 Mb of data.

Observing time needed: Each night, observing activities lasted approximately 8 hours per night, with 7 hours spent at the telescope, and 3 hours of observing the Pluto fields. After arriving at the telescope and opening the dome, a 1-2 hour wait was necessary to equalize temperature and improve seeing. There were 2-3 people observing at a time, but two people were sufficient after a routine was established.

MKO Data Analysis

Data description: The relevant portion of the sky was divided into 11 areas, and data were taken such that each strip fit almost exactly the boundaries of a single area. Areas I, III, V, VII, IX, and XI are abutting areas, and cover the portion of the sky in which Pluto is during 28 April - 9 June 1988. Areas II, IV, VI, VIII, and X overlap the adjacent areas. A group of network stars (VJ~J13-15) in this area was chosen, and coordinates were determined at Lowell. Each strip contained between 5 and 7 of these network stars. The star network of areas II and III, both of which contained the occultation star, were expanded to provide better fitting. These areas contained 60 - 70 stars each.

Data reduced: By 7 June 1988, the MIT team had reduced half of the data received: 2000 Mb. The reduction emphasized areas II and III because they had the greatest star density and because they both contained the occultation star. Pluto entered area III on 17 May, and entered area II on 25 May. Separate predictions were therefore possible from each area. The final area III prediction had 59 Pluto observations and 60 strips, and the final area II prediction had 15 Pluto observations and 36 strips.

Manpower: To reduce the 2000 Mb of data in six weeks required 2 people at 40 - 60 hours per week, and 2 people at 10 - 20 hours per week. There were 8 people working on data reduction at various times, both at MIT and MKO.

Compute Time: Extracting stars from strips for fitting (MKO):

30 minutes/strip on GC(generic computer)

Fitting for image centers (MIT-VAX-750):

Areas II and III: 4 hrs/strip

All other areas: 0.5 hrs/strip

Registering for predictions (MIT-VAX-750):

0.5 hrs/area

KAO Observing

Test Flight, 7 June 1988: During the KAO test flight on 7 June, we took high-speed series images of Pluto in preparation for the occultation, as well as astrometric frames to update the prediction. 32 frames were taken (1 minute exposures), centered on Pluto and the occultation tar. These frames had about 15 network stars per frame.

KAO Data Analysis

Image Fitting and Registration: One person worked on processing the astrometric frames through the data reduction pipeline to get star centers. The network stars were extracted from the strips using the GC (generic computer) which was set up in the front of the KAO aircraft. These extracted data boxes were then transferred via ethernet to the DEC VS2000 computer installed in the back of the KAO. The pipeline software had been installed on this VS2000, to allow it to do the image fitting and registrations. Two people started extracting the network stars from strips using the GC, after the KAO returned to Hawaii, at about 2 am local time. Another person worked the rest of the pipeline on the VS2000, and finished the reduction by 10 am the next morning. Each frame required 15 minute of processing time. Two people then worked on the registration of the strips to produce a prediction. This work required approximately 3 hours

Conclusion: The occultation of a star by Pluto was observed successfully. The discovery of an atmosphere on Pluto being a fundamental planetary science discovery.

Issues investigated

Can a telescience environment be defined that is useful in a full blown active and intense research environment? If useful, what are the principal parts of this environment?

Experiment Hypothesis

We hypothesized that a net increase in the groups scientific productivity could be obtained by the incorporation and use of enhanced connectivity, standardization, computational power, computational flexibility, and data storage methods.

Method of Investigation

1. Undertake a program to convince our group of the timeliness and applicability of the telescience concepts and methods to our environment.
2. Enhance our physical facilities with two microVAX II GPX computers added to our environment through Project ATHENA funding. This gives a firmer physical base of operation.
3. Identify the science experiment that would be the vehicle of this investigation.
4. Setup the optical disc archival program.
5. Plan details of the science experiment to capitalize on the new capabilities afforded by telescience.
6. Software standardization to be able to move programs easily from one computer to another.
7. Optimization of ethernet use for transfer of large files.

8. Coordinate this work with our collaborators at Lowell Observatory in Flagstaff Arizona.

Experiment results

1. We have successfully used many telescience concepts in our testbed.
2. Within our environment we have standardized machines and software to make the use of additional machines on our network very fast and efficient.
3. By standardizing on the UNIX, 'C', environment and as it worked out, DEC machines, these setup and use of additional machines on a very sort time scale for this type of operation became feasible.
4. The software was designed with archiving (see below) in mind. The use of fits headers and scripts allowed a pipeline approach to the data analysis and at the same time the capability of reproducing the data reduction if necessary.
5. The experiment used a distributed computational environment efficiently throughout most of the science experiment. Rapid transfer of image files to another workstation for further processing was a working reality during this experiment.
6. Archiving on optical disc has been performed efficiently and in a straightforward manner.
7. The discs are in fact stored in standard hanging files along with paperwork for the project. We believe this to be one of the most important aspects of our work as it provides a method to remove the data, correspondence, and programs from the on-line (magnetic disc) storage and thus ready the space for new work.
8. We have demonstrated that an upgrade in science capability can be obtained by incorporating telescience technology, methods and connectivity into our environment.
9. We have demonstrated an effective interface of our local (group) environment with a wider net, the MIT ATHENA project of many distributed workstations.
10. Finally we have done a remote observation (at Mauna Kea, Hawaii) and with the facilities at Cambridge Mass. been able to participate in that observing almost as if it was a local observatory.

(We still need a better form of data transfer than tapes by commercial air, [note-this corresponds to a bit rate of 20 kbits sec-1 averaged over 24 hours]).

3.5 Interactive Access to Purdue's Field Research Database (Purdue University)

The purpose of the experiment was to evaluate the effectiveness of making Purdue's Vegetation & Soils Field Research Database available for interactive access by local and remote earth science users.

Experiment Description and Summary Conclusion

The vegetation and soils field research database at Purdue includes nearly 300,000 spectrometer and multiband radiometer observations for 200 vegetation and soils experiments collected from 1972 to the present. The spectral data are supplemented by an extensive set of biophysical and meteorological data acquired for each experiment. The majority of this data have been collected as a part of NASA grants and contracts. These data have (until this project) been stored on 49 magnetic tapes accessible only from an IBM mainframe computer running VM-CMS with locally developed software. At least in part because the data were relatively difficult to access, the data were not frequently used. As a result, the tapes were stored in the basement of the library, adding to the difficulty of accessing them. They had to be checked in and out the main computing center when researchers wanted to access the data.

For this experiment, a database management system and a workstation were selected to use to manage the catalog of the field research data and at least of subset of the database itself. The workstation was connected to Purdue's network of computers which in turn is connected to the Internet and Bitnet. Also a modem and a phone line were connected to the workstation for remote access via dialup. The user interface to the data summary was prototyped using Apple's HyperCard.

A catalog of the database was put into the database management system. Also the capability was implemented in the database management system to allow users to view and transfer copies of the database files that were stored on the workstation hard disk. Comparisons were made of the effectiveness of interactively accessing the database from local and remote sites.

Accessing the data stored in the basement of the library required 1 to 2 days to find and read the experiment descriptions (book form), get the tape, carry to the computing center and then run the software on the main computer to copy the data into a disk file. Accessing the database from the new on-line system from local sites is very effective. Researchers at Purdue can log onto the system, select the experiment data that they want and get a copy of that data (if it is on the system) within 15 to 30 minutes. Accessing the database from remote sites via the Internet takes longer and can be frustrating because of the slow interactive response times. The effectiveness of interactive access via the Internet is related to the distance from Purdue. In general the best approach from remote sites is to dial in to the Purdue workstation to browse the database and then use the Internet to transfer the desired data.

Issue Investigated

The issue investigated was to determine if one can take advantage of current computer networks and communication capabilities to make accessing remote databases for researchers easier and more productive.

Experiment Hypothesis

The hypothesis was that using off the shelf DBMS to manage the database on a workstation connected to the computer networks would make it easier for a researcher to evaluate the data available, determine the data to be used for a study, and obtain a copy of the data. The capability would also make life easier at Purdue by tying up fewer resources in making copies of tapes and mailing them to researchers.

Method of Investigation

Oracle was selected as the RDBMS. Other DBMS's evaluated were Ingress, Unify, and Sybase. Oracle was selected because, given the need to be Sun compatible, it provided the best compromise between capability and cost to the university. A Sun 3/60 with 12 megabytes of memory, a 141 megabyte hard disk and a cartridge tape drive was selected as the workstation. The Sun workstation was installed and connected to the Internet in late February 1988. Oracle was installed on the workstation during March 1988.

US Robotics Courier HST 9600 baud dialup modems were obtained to use for testing high-speed dialup up access to the database on the workstation. The summary (or catalog) of the field research database was designed consisting of eleven files. The catalog summarizes 200 experiments with 240 spectral instrument-sets of data and 370 instrument-versions. (More than one spectral instrument were used for some experiments and the data have been processed in two version for some experiments.) Before the workstation arrived, a prototype of the spectral database catalog was developed using Apple's HyperCard to verify the catalog files and test different methods of displaying information to users. This effort proved to be very useful.

Software was developed to convert the database stored on the magnetic tapes in EBCDIC character, integer and IBM floating point formats to ASCII character formatted disk files that could be stored on the workstation hard disk and Write-Once-Read-Many (WORM) optical disks. The decision was made to convert all measurement to character format so that it would be much easier for researchers to view the data in the database.

After the Sun and Oracle were installed, the field research database catalog files were loaded into Oracle and the user interface screens were implemented along the lines of those developed using HyperCard. The capability was implemented within the Oracle database summary to allow the user of the system to transfer the data and descriptive files for selected experiments to his/her computer system via FTP across the Internet or Kermit across dialup links. As of mid-October 1988, 165 of the higher priority instrument-sets of the database have been loaded onto the hard disk on workstation. The entire database will not fit on the hard disk. The plan is to have the entire database stored on optical disk so that those files not on the workstation hard disk can be moved to the hard disk when needed.

A dedicated phone line and a US Robotics modem have been recently connected to the Sun workstation (early October 1988).

Users have accessed the database on the workstation from the University of California at Santa Barbara, Kansas State University, NASA Ames Research Center and NASA Goddard Space Flight Center as well as from Purdue University.

Experiment Results

Having the database available on the workstation has been very helpful. It is especially useful for those researchers at Purdue University. The system is very responsive for interactive searches and data transfers. The transfer rate from computer to computer at Purdue via FTP on ethernet has been around 50 to 70 megabytes per hour. Transfer from a Macintosh personal computer to the Sun workstation via AppleTalk, Kinetics box, and ethernet has been 10-12 megabytes per hour.

Interactive access to the database from remote users across the country via the Internet has not been as successful because of the load on the Internet. In general the closer the remote site is to Purdue, the better the interactive access is. A researcher from Kansas State University was able to transfer data at 1.3 megabytes per hour across the Internet. There were significant pauses, 10-20 seconds, during the interactive session with the Oracle database waiting for the screen to refresh with a new interface display form. Experience from UCSB, Goddard, and Ames to Purdue for interactive access included some waits of up to 30-50 seconds for a screen update.

An observation in support of the Internet is that, even with the long waits, one is able to access data and transfer it much more quickly than sending a letter or making a phone call to request the data, wait for the host site to make a copy and send through the mail. The difference is minutes-hours compared to days. However, the frustration factor is that the user feels like he/she is wasting a lot of time waiting for the screen to update or starting a session over again if the link goes down in the middle of a session. Also there is a tendency to strike keys trying to get the system to respond which get stored in the queue and cause problems with the session when they get executed. The overall conclusion is that consistent waits of more than 10-15 seconds for a screen update for an interactive session will cause the researcher to try other approaches to browse the data.

3.6 Access to the University of Wisconsin's Man-Computer Data Access System (McIDAS) (Purdue University)

The purpose of this experiment was to evaluate and compare the effectiveness of high-speed dialup and Internet access to the University of Wisconsin's McIDAS system from Purdue University in West Lafayette, IN.

Experiment Description and Summary Conclusion

This experiment involved accessing AVHRR (Advanced Very High Resolution Radiometer) data collected by one of the NOAA satellite in near real time to be used for drought studies in Indiana. Arrangements were made to use the University of Wisconsin's

McIDAS system. McIDAS is the system that scientists at the University of Wisconsin have developed to allow researchers to access and view meteorological data. See the discussion of the University of Wisconsin's experiment for more information on McIDAS.

We were able to access the AVHRR data within 27 hours after it was collected using the McIDAS system and the Internet. This compares with one to two weeks or more for getting copies of the data through the mail.

Issue Investigated

The issue investigated during this experiment was "can McIDAS and the Internet be used to access near real time satellite data for research?".

Experiment Hypothesis

By using the capability of the Internet and McIDAS, we hypothesized that we could get a copy of AVHRR data much sooner than we could by requesting the data via a phone call and have a tape mailed to us.

Method of Investigation

Researchers at Purdue had previously used some frames of AVHRR images to help access the effects of the drought in Indiana. However, the data were several days (5 to 14) old before they could get a copy of it. We were interested in finding other ways to obtain near real-time data, i.e. only hours old data

Researchers at the University of Wisconsin indicated that during the summer of 1988 they had implemented the capability to ingest the data from the NOAA-9 and NOAA-10 spacecraft which includes the AVHRR data. AVHRR data is presently not a standard McIDAS product, but Ralph Dedecker indicated that they could run some test cases to see what was involved in handling AVHRR data. A test case was arranged for mid September.

A clear AVHRR image was collected over Indiana on 9-14-88. Personnel at the University of Wisconsin's Space Science and Engineering Center ingested the data from their DOMSAT link and then transferred the visible and near infrared channel images to their PC-AT McIDAS computer. We were notified the next day that the data was available. We logged into their PC-AT McIDAS computer using FTP and transferred 1.2 megabytes of data to a Sun workstation in 8 minutes. Twenty-seven hours after the data were collected, the AVHRR image was up on a display station being evaluated.

Experiment Results

The experiment demonstrated that satellite data can be transferred to researchers to use in near real time. Near real time, of course is a relative term. The comparison for this experiment was with obtaining a copy of the data through the mail. The approach used in this experiment reduced the delay time from 7-14 days to about 1 day which for us is significant

We are presently working with the University of Wisconsin to determine what resources would be required to access the AVHRR data routinely for future work.

3.7 Spacelab-2 Data Flow Simulation (Stanford)

The primary goal of the Stanford University Telescience effort is to evaluate delivery of real-time experiment data to a remote site. Both conventional data circuits and satellite data broadcast are being evaluated for their efficiency and cost effectiveness. The primary data being transmitted are the science and engineering data collected from the Vehicle Charging and Potential Experiment (VCAP) during the Spacelab-2 shuttle mission. The main purpose of the experiment is to investigate the difficulties of establishing data circuits for real-time data, and the requirements for the software to capture and reliably transfer the data.

Experiment Hypothesis

The principle task of the Spacelab-2 data flow simulation is the installation of several data links from the Data System Technology Laboratory at Goddard Space Flight Center to the SUNSTAR Laboratory at Stanford University and the development of software to provide collection and reliable distribution of real-time experiment data. A variety of data links are being used to evaluate the efficiency of real-time data transfer between Goddard and Stanford. These links include a 56 kbps dedicated data link running the TCP/IP protocol, and a satellite data broadcast link.

The satellite data transmission project is intended to demonstrate a method of distributing real-time data at high rates to multiple sites. The basic structure of the satellite communications system is a central transmitting facility which relays data received from space based instruments. These data would be sent out over satellite links in a broadcast mode to receive only stations at remote sites. Low speed duplex links between the remote sites and the central transmitting facility would provide for coordinated computer control of the data transfer and retransmission of lost or corrupted data. A broadcast satellite data distribution network is less expensive than multiple duplex transmission capabilities, but the separation of the control and retransmission link from the main data link must be proven to be viable.

Investigation Method

At Stanford, a VAX 11/750 computer and a VAXstation II workstation are being used for the testing and development of the satellite communication hardware and software. At Goddard, VAX 11/750 and VAX 11/785 computers are being used to capture the Spacelab-2 VCAP and Orbiter status data being played back from the original data tapes over a digital data link to the Goddard Spacelab Data Processing Facility. Software for the transfer of data between the computers has been developed at both institutions. All these computer systems are currently running the VMS operating system.

Several digital communication links have been established between the Data Systems Technology Laboratory at Goddard Space Flight Center and SUNSTAR Laboratory at Stanford University. A dual 9.6 kbps link using the DECnet protocol between the computers at Goddard and Stanford has been operational for several years as part of the Space Physics Analysis Network (SPAN). These DECnet links can be used for the coordination of the satellite data transmission, and are used for day to day communication. A 56 kbps data circuit has recently been installed between the Stanford and Goddard computers. Bridge

Communications network gateways at both institutions provide a 56 kbps interface using the TCP/IP protocol between the computers on the Goddard ethernet and computers on the Stanford ethernet. Exelan communication boards provide a TCP/IP ethernet interface for the computers at Goddard, while the Multinet software is used at Stanford to provide both DECnet and TCP/IP ethernet access.

A simplex satellite data link at 80 kbps is being completed between Goddard and Stanford. The installation of the satellite broadcast communication link has taken longer than anticipated primarily because of difficulties in providing a satellite transmit capability for the Data Systems Technology Laboratory at Goddard. The receive antenna at Stanford University has been installed, but needs to be adjusted for the assigned communications satellite and frequency.

The interfaces for the digital data link of the satellite data broadcast system have been completed except for the final integration into the data transfer software. The Advanced Computer Communications (ACC) company 5000 and 6000 series of computer interface boards are being used to connect the computer to satellite modems. The interface boards were initially run using the full HDLC protocol software in order to insure that the boards worked and to begin development of the interface communications software. After this initial testing, driver software modified by ACC was installed in the communications boards in order to allow the data link to operate in simplex mode. The communication boards were interfaced to the satellite modem equipment, and test data were successfully transferred at 80 kbps between the two computers with the satellite modems in the data path. The integration of the communication board software into the data transfer software is nearly complete. After the communications hardware and software have been satisfactorily tested in the laboratory, the electronics units will be connected to the satellite dishes and the host computers.

The data transfer software has been installed and successfully run between computers locally at both Stanford University and Goddard Space Flight Center. Most aspects of the transfer software have been tested between the two sites over both the low speed DECnet links and the higher speed TCP/IP links. The testing of the data transfer software through the satellite modem equipment is proceeding in parallel with the completion of the satellite data link.

One additional aspect of the simulation of the Spacelab-2 mission is the development of a workstation based display of the Shuttle Orbiter to indicate both predicted and realtime parameters important to the experiments being conducted. This model has been used for planning realtime operations of the Vehicle Charging and Potential Experiment on Spacelab-2 mission, and for analysis of data after the mission on a Silicon Graphics IRIS workstation. The display shows the electron trajectory emitted by a low power electron source and vectors indicating the direction to the sun and the center of the earth, the earth's magnetic field direction, and the direction of motion of the Orbiter. The original program has been converted from the Fortran programming language to the C language and has been re-organized in a more structured form. In particular, the ability to read parameters from realtime and remote data sources has been incorporated in the program, and one particular aspect of the remote data access has been to access display parameters from another

computer across an ethernet interface. The modified code has been tested by accessing display parameters from remote workstations running under both the Unix and VMS operating system through a standard ethernet interface.

Results

Installing and maintaining data communication links is very time consuming and is plagued by a variety of small problems. These constraints will likely tax the resources of most small research groups, and most often data links will need to be established as institution based resources. Unfortunately shared data links are not always reliable for real-time monitoring and control of remote systems. A satellite broadcast mode of transferring data has possibilities, and the joint work between Stanford University and Goddard Space Flight Center is close to the actual testing phase.

Initial testing and software development for the satellite data broadcast system indicated that the interface boards worked and allowed transfer of test data files between the two computers. Initial tests verified the operation of the ACC communication boards using the simplex mode between a VAXstation II workstation and a VAX 11/750. The interfaces were able to handle burst mode data rates of 2 Mbps, but the writing of the received data onto disk storage reduced the continuous data transfer rate by an order of magnitude. More efficient data storage algorithms exist to speed up the data storage rate, but real-time data storage in typically available computer systems is a problem for moderately high data rates.

3.8 Effectiveness of Optical Disk Media for Database Storage (Purdue University)

The purpose of this experiment was to examine the usefulness of storing the field research database on a Write-Once-Read-Many (WORM) optical disk for archive purposes.

Experiment Description and Summary Conclusion

Optical disk media have the capability to reduce the space requirements and access time to large databases. Purdue's field research database is stored on 49 9-track magnetic tapes and consists of 500-600 megabytes of data. A WORM drive was selected that could be attached to a Macintosh II personal computer. The tapes were copied as is (no changes in data format were made) to a disk on the IBM mainframe (3090) and transferred via FTP across the ethernet and AppleTalk to a hard disk on a Macintosh II. The tape file copies ranged from 1 to 36 megabytes in size.

Also the data were converted to ASCII formatted files that can be easily interpreted by researchers and stored on the optical disk. These files can be considered the working database files.

Around half of the tapes have been copied to optical disk. We have been pleased with our experience so far with this project. We feel that we have a much better control of the data and can get at it much more quickly. We should be able to get all 49 tapes stored on one 800 megabyte optical disk.

Issue Investigated

The issue investigated was to determine if access and management of 500 megabyte database could be improved by copying the data to a 800 megabyte WORM optical disk.

Experiment Hypothesis

The hypothesis was that storing the database on an optical disk that would replace 49 tapes would make management and access to the data much easier and quicker. One would not have to depend on other groups (the library) for access to the data. The data would not be so heavy to carry around (49 tapes) and it would be much easier and cheaper to make back-up copies. Also researchers could make copies of the data much easier from the optical disk media.

Method of Investigation

WORM optical disk drives were evaluated for the Sun 3/60 workstation and the Macintosh II personal computer. A LoDown 800 megabytes optical disk drive was selected. LoDown (now Laser Optical Technology) had disk drives for the IBM PC and were working on one for the Sun workstation. A WORM drive was not selected for the Sun workstation because the available WORM optical disk drives for the Sun were expensive and required that we write software to do the file transfer.

The decision was made to put two versions of the database on optical disk. The first version would be the archive version - exact copies of the tapes. We wanted to be certain that we had an original copy of the data, in case it was found later that there were any conversion errors. The second version would be organized in experiment and instrument files that would be easier for researchers to access. The data on tape were generally organized by instrument and the date the data were collected.

A few tapes at a time were checked out of the library and checked into the Purdue computing center. The tape was copied to a disk file on an IBM 3090 and then transferred to an 160 megabyte disk on a Macintosh II hard disk across ethernet and AppleTalk using FTP. After the 160 megabyte disk was nearly full, then the 160 megabyte disk was copied to the optical disk.

Experiment Results

The experiment was delayed several months waiting on upgraded software for the WORM drive that would allow one to copy individual files to the optical disk. We have not received that software yet, but decided to proceed by copying a volume (entire hard disk or floppy disk) at a time whether full or not. This causes the storage on the optical disk to be less efficient. Delivery of the upgraded software is "soon".

The transfer rate from the hard disk to the optical disk is around 25 megabytes per hour. The transfer rate from the optical disk to the hard disk is around 45 megabytes per hour.

To date, 20 of the 49 tapes have been copied to an optical disk. Some of the ASCII formatted files have also been copied to the optical disk.

3.9 Remote Data analysis for EUVE (UC Berkeley)

One of the important components of the EUVE project is the Guest Observer (GO) program for spectroscopic observation with the Deep Survey/Spectrometer instrument. The current plans for the GO data analysis include, a local database and distributed scientific users having various level of access to the database. These components are all similar to those in the Browse system, developed at the University of California at Santa Barbara (UCSB). Furthermore, the EUVE data makes heavy use of images of astronomical objects, another similarity with the Browse system. Finally, the spectroscopic data are obtained by imaging spectrometers, and so they can be treated like images. For all these reasons we have explored the possibility of incorporating the electronic Browse system or some of its key components in the EUVE GO data analysis system. The details of the investigation is described in Chakrabarti et al., 1988c.

Experiment Description and Summary

For this study, we have examined the use of Browse for the EUVE GO data analysis program. Several features of the Browse system can be incorporated for the GO data analysis program as described below.

At the discretion of the guest observer, the off-line mode of operations will require permanent files that are generated in the production processing. Binned photon and exposure maps are generated that may be deconvolved for some studies (e.g., extended sources and crowded fields), while detailed photon and exposure data are saved in "pigeon hole" files for studies of individual sources. It is expected that the latter data set will be more commonly used. The collection of detailed data into pigeon holes (as applied to spectra) is governed by the list of sources detected in the deep survey instrument which provides a direct image of the target region. These data will be similar in format to those in the Browse database. As with the survey data sources may be found which were not expected or which may be brighter than expected and thereby contribute to noise in the spectrum. Thus, off-line analyses are required to collect data for sources that were not in the original catalog. Data for newly discovered sources will be collected from prior observations for future analysis.

The main differences between the handling of survey and spectroscopic data are in the treatment after collection into a permanent database. Generally, all data are handled by the GO interface software which allows data and software transfers via phone lines, networks, or physical media (e.g., optical disks for large requests or printout for users without access to computer networks). Browse has schemes that account for the different levels of access to the database; even different types of user environment (e.g., dumb terminal vs. graphics workstation interfaces are also included in their interface. We will examine these schemes before selecting the GO interface.

The GO Interface selects the applicable portion of the database to minimize file transfer time and should warn the remote user if the requested data are not in the current database. In such a case, a special request is generated that requires action by the database manager, which controls the archive data. Requests for unavailable data are handled by generating a set of new coordinates for guiding the selector program via the database manager. This process insures that the raw data are managed properly by a on-site operator of the database managing program. Unlike Browse, we plan to allow remote logins but access to the production network will be restricted for security. Temporary files may be created by running remote processes and then transferred to the GO's home institution.

The GO database plan calls for a file called history. It contains general orientation information, instrument house keeping information and other parameters such as sun angle positions of the moon and other planets, etc. There are also plans for storing intermediate results such as exposure information, photon maps, data quality indicators, etc. This database is therefore essentially the same as catalog in the Browse system, which contains information that describes the quality of the data and other attributes.

It is expected that a number of specialized I/O and data analysis packages will be written specifically for use in the IRAF data analysis environment, an image analysis package developed by the National Optical Astronomy Observatories (NOAO). These packages will be available for remote execution or transfer to the GO's home institution for installation in IRAF. This is similar in concept to the software Browse will provide to its remote users for manipulation of images. The packages supplied for EUVE data analysis will run on any hardware configuration to which IRAF has been ported. Using IRAF's potential networking functions, one may also invoke this package at the remote site without transferring the applicable files.

We also anticipate several simultaneous observations with EUVE by other instruments (ground- or space-based). These GO nodes will contain special data not in the EUVE database as well as specific scientific interest and expertise. These characteristics are also incorporated in Browse and are included in the directory and user databases. We anticipate that the EUVE GO database management plan will include similar features.

Issues Investigated

We investigated the usefulness of electronic browse for the EUVE guest observer data analysis program. The Browse system developed at the UCSB was examined in detail to check that the major EUVE GO functions can be supported. The incorporation of IRAF image analysis system in Browse was also explored.

Experiment Hypothesis

We hypothesized that the Browse system can be easily adopted for the EUVE GO program.

Method of Investigation

We visited the University of Santa Barbara and took part in a demonstration of the Browse system operating on a wide variety of remote sensing data. We also were briefed on its specific capabilities. At Berkeley, we had several meetings defining the EUVE GO data analysis requirements. A comparative study was undertaken to determine the adoptability of Browse for the EUVE system.

Experiment Results

We are very encouraged by the capabilities of the Browse system which should be adaptable for the EUVE image and spectroscopic data. The image database and the astronomical catalogs to be used in the EUVE program is similar to the remote sensing data obtained from a variety of platforms. We feel that Browse is a potential candidate for the GO data analysis system.

Unfortunately Browse, as it is presently implemented, uses a VMS operating system. All of the EUVE software is built around a network of Sun workstations running UNIX. Our SC scheme, described earlier makes heavy use of UNIX facilities. Therefore, we will not be able to incorporate Browse readily.

Due to financial and schedule constraints we have not been able to use actual astronomical data and used the Browse system to analyze them from different remote locations. We also have not explored the availability of Browse for use at the SSL, UCB without regular supervision by the UCSB personnel.

3.10 Real-time Rocket Data (UC Berkeley)

Our final experiment used real-time telemetered data from sounding rocket. The sounding rocket, called Berkeley EU Airglow Rocket Spectrometer (BEARS), carried a 92 lbs spectroscopic payload to 963 km altitude on September 30, 1988. The reduced telemetered data was sent to the SSL from the Wallops Island using a 9600 baud modem over a telephone line. During the flight, scientists at Berkeley communicated over telephones with their counterparts at the Wallop Island regarding the instrument and its operation.

Experiment Description and Summary

The BEARS experiment and its scientific objectives are described in detail in Chakrabarti et al., 1988d. The pay load consists of (1) a high resolution (approximately 0. angstrom) spectrometer to measure the EUV emissions (980-1360 angstrom) of the Earth's dayglow, (2) a moderate resolution (15 angstrom) EUV spectrometer (250-1450 angstrom) to measure the solar irradiation responsible for the photoelectron production, and (3) a hydrogen Lyman Alpha photometer to monitor the solar irradiance and geocoronal emissions. Although at the time of the writing of this paper we started our teleanalysis plans, they were not mature enough for inclusion in this paper. We hope to report these results in an upcoming publication.

Issues Investigated

The primary goal of the experiment was to involve scientist at their home institution in the interpretation of real-time data. To our knowledge this was the first time a telemetry stream was decommutated in real time and spectra from the instrument, not just monitor values, were transmitted to remote locations for analysis. The final goal of this experiment is the development of hardware and software systems for real-time teleanalysis in conjunction with teleoperations.

Experiment Hypothesis

The primary hypotheses of the teleanalysis experiment are Real-time teleanalysis even for short duration sounding rocket flights are useful from their scientific return. The technology is mature enough to support teleanalysis.

Method of Investigation

An IBM PC/AT based system equipped with special hardware was used to collect selected words from the telemetry stream. Specially written software used the data words thus obtained and formed spectra from the three spectrometers flown aboard BEARS. This was rather computer intensive, since the detectors employed were position sensitive detectors, and telemetered three 14-bit words describing the locations of each photon event recorded by the detectors.

The spectra generated were transmitted to the SSL from the Wallops Island, using a pair of Microcom 9600 baud modems using the MNP protocol for data compression and error correction. The spectra were displayed to the SSL real-time data analysis team. Intensity ratios of selected spectra lines, background rates, spectral contents were all evaluated. For this experiment, we did not have any commanding capabilities. So the analysis were used to interpret the quality of the collected data and problems with the instrument.

Experiment Results

The teleanalysis experiment showed how valuable real-time teleanalysis can be. We did discover a problem with the instrument. Had we any commanding capability, we would have tried to interact with the instrument, to fix it. This is one of the prime goals of telescience - to give scientist an environment, which "feels" laboratory-like. The reduction of telemetered words into physical form (intensity vs wavelength) and their representation in familiar graphical form made the task of real-time interpretation a success.

3.11 Browse Testbed (Univ. of California, Santa Barbara)

The objective of this task is to evaluate issues of science data management, specifically by attempting to identify a subset of the components needed between a scientist's workstation and remote archives of digital spatial data.

Experiment Description and Summary Conclusion:

A software testbed was developed, incorporating a prototype suite of tools to be able to examine and evaluate image data by both attribute and content. A diverse user group is using the testbed, and their experiences and reactions to this prototype will form a primary part of our final recommendations to TTPP.

Issues Investigated:

- Can a common suite of tools be identified that provide scientists in a variety of disciplines appropriate information to make a value judgment about the utility of a dataset?
- Are in-place networks sufficient to support this kind of function?
- Can a limited set of pre-processing functions provide browse image datasets for a diverse user community, thus making the on line storage of the full-resolution raw science data unnecessary?

Experiment Hypothesis:

A small subset of image attributes, processed browse image datasets, and graphic query functions can provide a inexpensive, useful environment to evaluate the utility of multispectral data stored in a remote archive.

Method of Investigation:

The testbed software was written in Fortran 77, using the RI database manager and the GKS device-independent graphics library. The system is menu driven, and supports catalog an inventory queries. The specification of a geographic region of interest is either menu-driven, for those with character-base terminals, or based on an electronic map display, which uses a Tektronix 4010-like graphic terminal. 4010 terminal emulation is extremely common, particularly with the recent release of Kermit, in the public domain, which supports this terminal through emulation on most IBM PC-compatible graphics displays. Dial-access, SPAN, and Internet access to our testbed are all operational.

Once a potential image dataset is located, appropriate file transfer capabilities (i.e., Kermit for dial access, DECNET copy, or Internet ftp) permit the user to download the relevant processed browse files to the local workstation for further analysis and evaluation. We also provide an image display and enhancement program to interested collaborators for this last function, for the IBM PC-EGA platform. We are porting the testbed software from VAX to the PC in the next fiscal quarter, partly to begin to exercise the problem of query of distributed spatial data archives.

Experiment Results:

Build II of the testbed is operational, and our user team is evaluating the system at this time. We have insufficient user sessions with the new software build at this date to identify trends. We observe that the Internet is not a reliable means to support remote login to the

testbed when using the electronic map module, although file transfers through the internet have been reliable. The USR and Telebit high speed modems have been surprisingly effective. User response to the revised electronic map module has been positive, and we believe this may be a generally useful user interface mechanism. While 9600 BPS system access is acceptable for the menu query functions and the electronic map, it appears unacceptable by a factor of 2 to 20 for the transfer of browse images.

3.12 Telescience Field Experiment (Univ. of California, Santa Barbara)

Our objective was to attempt a campaign-style field experiment, to exercise existing telescience capabilities.

Experiment Description and Summary

The specific science problem was to estimate rates of evapotranspiration in a savanna ecosystem, as a component of a research program into the interplay between vegetation and hydrology. Overall, this experiment clearly indicated the value of a telescience approach to our future field campaigns, and specifically illuminates areas where more work is needed.

Issue Investigated

Are current capabilities in terms of remote access to distributed databases and processing systems sufficient to support short term multidisciplinary field campaigns?

Experiment Hypothesis

An ordinary personal computer and high-speed dial-up modem at a remote field site are sufficient tools to conduct this kind of campaign experiment.

Method of Investigation

The telescience field experiment was conducted at a U.S. Forest Service experiment station, a savanna ecosystem at Coarsegold, California. We attempted to coordinate our field measurements with a Landsat 5 overpass, SME data acquisition, and an aircraft mission for photography and digital scanner data. Field data collection included sampling for percent canopy cover and ground surface brightness temperatures. Only a limited amount of the range station could be sampled with the limited time and staff available, so area samples of percent canopy cover were regressed against video digitized photography. This imagery was located in the UCSB BROWSE testbed, accessed through dial-up modem from the field.

The image data was processed with software on a separate computer system at UCSB, also controlled from the field. This processing included rectification and enhancement. The resulting images were downloaded to the field computer and displayed using our locally-developed software. Statistical analysis of the field data were analyzed while still in the field, again using dial-up access to campus minicomputers. Had real-time imagery been available, the statistical analyses could have been used to verify the model calibrations with field data on-site. Further, we accessed the McIDAS system from the field site, and the GOES imagery available in this manner could have been used to coordinate future real-time image data acquisition with incoming weather systems.

Continuing analyses include calibrating the Thematic Mapper scene acquired during the field campaign, and understanding the applicability of the SME datasets to our models.

Experiment Results:

Results and Conclusions

Logistical and systems problems prevented us from collecting all the scientific data we had planned. Remote access to the databases and processing systems at UCSB, Purdue, and U. Wisconsin was possible from the field. We now better appreciate the problems and possibilities of real-time remote sensing measurements, based on our experience working with University of Colorado in trying to set up contemporary SME observations. We are now preparing detailed plans for our next field season at the site, based on these experiences.

3.13 Variable Resolution Transmission of Digital Image Data (Univ. of Rhode Island)

The University of Rhode Island has developed and tested interface software ultimately intended for addition to image processing systems. The software implements user-selectable, variable resolution transmission of digital image data over network and point-to-point communications links. It is intended to make it feasible for users to access and visually browse a remote digital image archive.

Experiment Description and Summary Conclusion

Because remotely sensed images can be corrupted by data drop-out, can have obscuring features such as clouds, or can simply contain no feature of interest for some particular investigation, making use of them typically involves, as a first step, visually searching (browsing) through sets of images to find those with information relevant and usable for an application. Having purely symbolic descriptions of an image or image set (such as might be contained in a catalog) is usually not sufficient to locate relevant images. For example, a descriptor in a catalog that noted that a particular image was 75% cloud-obscured would be of no help if the investigator was interested in the cloud-free sub-areas of the image.

On-site browsing of images with appropriate equipment is a tedious but tolerable chore. Accessing an image takes on the order of seconds. On the other hand, the large volume of data in even a single image and the relatively low capacity of standard transmission channels makes remote browsing of image archives impractical because the transmission time is too long. The theoretical lower limit on transmitting a single 512 x 512 image of 8-bit pixels at 9600 bps is about five minutes. Thus, given existing software for image transmission and typically available data rates, it is a practical impossibility to visually browse a remote archive.

In our project/experiment, we investigated methods of reducing the total amount of information transmitted per image while still providing the user with the information necessary for the task. The methods involve transmitting the image in a sequence of increasing resolution representations of the original and of allowing the user to select a sub-

area of the image for increasing resolution presentation while holding the remainder--the context or background--fixed at a lower resolution. Reducing the total amount of transmitted information reduces transmission times and makes it feasible to browse a remote archive.

Issue Investigated

Is it possible to provide remote users with effective visual-browse access to a digital image archives over modest bandwidth (<= 9600 bps) transmission lines.

Experiment Hypothesis

We hypothesized that software implementing the variable resolution transmission techniques would be a satisfactory means of enabling remote access to a remote archive of AVHRR images.

Method of Investigation

In very general terms, our method was essentially software development, testing, and evaluation. One of our M.S. students, Eugene Tsai, designed and wrote programs implementing the variable resolution approach for an IBM PS/II 50 with VGR graphics. These programs were designed to allow dial-up connection to an archive (such as ours at URI or one at the University of Miami) running the DSP image processing system. Somewhat later we hired a half-time programmer, James Gallagher, to implement a similar system, this one however differing in that it was designed to allow a SPAN node running the DSP system to access images archived on another SPAN node running DSP and using SPAN as the communications channel.

The primary implementational basis for the systems was the image pyramid, an ordered series of reduced resolution representations of an original in which, moving up from the base, each representation has half the resolution of the one below. In the pyramid each pixel at a given level is a function of the four pixels immediately "beneath" it. Various functions, e.g., mean, mode, etc., can be used to generate the pyramid. We found that a simple sampling procedure which uses the value of a pixel from a fixed quadrant to represent the value of the quartet was satisfactory for browsing AVHRR images. This meant that the pyramid generating algorithm can use only array indexing arithmetic and hence is very fast. It also meant that only array arithmetic is needed at the receiving end to decode and that even if the entire pyramid is transmitted no more data than was contained in the original need be sent.

These systems went through several revisions and were tested in a number of settings. The IBM system was used as a demonstration system at a TTPP meeting in Boulder at which we effectively browsed our own archive using a 1200 baud modem. Tsai also used the system in an extensive test in which his goal was to remotely select, from a set of over three hundred images, those which were cloud-free in the area of a set of moorings. The DSP systems developed by Gallagher were installed here at URI, at the University of Miami, and at Goddard Space Flight Center and were used in a series of tests and demonstrations.

Results

Tsai's test mentioned above is the primary basis for the results and evaluations reported here. Tsai, while browsing AVHRR images with the intent of selecting those useful for calibration purposes, found that he was able to make a decision about an image in every case by level seven (effectively, a 128 x 128 reduced resolution representation of the original 512 x 512 or level 9 image) and in over half the cases by level 5 (32 x 32 reduced resolution representation). The mean time to browse an image at 1200 baud in his tests was approximately 20 seconds (versus 18.2 minutes to transmit the entire image). Though we do not have hard data to point to at this time, twenty seconds seems well within the patience of most users. Thus, even at 1200 baud, these techniques can be used effectively for some browsing tasks. Tsai's program was not designed, nor did we have the equipment to test, these techniques using 9600bps modems; but given other results reported by TPP participants using 9600bps modems, the range of usefulness of the system should be easily expanded.

On the other hand, results with the system using SPAN as the communication link was less positive, though not disheartening. The problem with SPAN is the effective transmission rate. Although we are connected to Goddard via a 9600 bps line and thence to the rest of the net, under some conditions of network load, it is possible to have an effective transmission rate significantly less than 1200 baud making browsing more difficult and tedious. Since the point at which browsing becomes untenable varies from person to person and also with the purpose or intent of the user, we are reluctant to categorically announce a lower bound on the minimum effective transmission rate necessary for remote browsing, but 1200 baud would seem reasonable. With standard compression techniques (difference, Huffman codes), we think we can achieve better than minimum effective rates even under adverse network load, but have not yet added this to our systems.

3.14 Delivery of Real-time Meteorological Products (Univ. of Wisconsin)

The University of Wisconsin-Madison has developed a bridge to NSFnet and provided MODEM ports for transferring real time meteorological data from the McIDAS computer system to remote earth science users.

Prototyping at the University of Wisconsin produced results suitable for either teleoperations or teleanalysis. Teleoperations aspects were addressed in cases where meteorological products were used for directing procedures during field operations. UCSB applied the data in this way by utilizing GOES cloud images for directing operations in a field experiment (see below). The meteorological products also provide input data sets useful for merging with other data sources for deriving post data collection results (Teleanalysis). Purdue used AVHRR data products in this manner.

Experiment Description and Summary Conclusion

Rapid access to meteorological information is frequently required during Earth science experiments. Real time information can be used for directing experimental activities and logged meteorological information is often useful for post data collection analysis. The

University of Wisconsin Space Science and Engineering Center (UW-SSEC) has been a leader in satellite meteorology for nearly two decades. SSEC has developed both remote sensing systems for data collection and ground processing, and meteorological database systems. The McIDAS (Man Computer Data Access System) was developed at the UW-SSEC and is currently installed and operating. It serves as a platform and real time database system for various research efforts and as a model for similar operational systems installed world wide. Use of the McIDAS database has previously been limited to users who had direct access to the system for their meteorological investigations. The Telescience experiment at UW-SSEC centered on making McIDAS information available to a wider field of scientists.

Work on the Telescience Program provided the link to allow real time McIDAS products to be transferred to selected earth science users via dial up modem connections and via the TCP/IP network. Only preliminary evaluation of the products has occurred during the Telescience program. While Purdue University has recognized the utility of some of these products (AVHRR) and UCSB has collected and displayed real time GOES products during field experiment operations that provided the TTPP with initial results, an insufficient user base has developed to draw definitive conclusions as to McIDAS product utility for general earth science programs.

Issues Investigated

Two basic questions were examined during the experiment. First, could McIDAS products be made available to remote users via standard communications schemes in a timely enough manner to still be considered "real time". Second, an examination of the utility of McIDAS products was considered.

Experiment Hypothesis

By providing a mechanism for transferring real time McIDAS data sets to remote earth science users, it was hypothesized that earth science research and experimentation could be augmented with the additional data, capable of improving resultant information derived from this research and for directing experimental operations.

Method of Investigation

IBM PC based software was developed and distributed to allow earth science users (Purdue, UCSB, and Colorado) to acquire McIDAS products via a standard MODEM connection and to display GOES imagery on a scheduled basis. A bridge was developed and implemented that allows TCP/IP network users to acquire these same McIDAS products.

Because of the unique nature of the real time data, data volume, and method of implementation, access to McIDAS and its database has been limited to custom hardware configurations and protocols. During the Telescience program, McIDAS data has been delivered to earth science users using standard telephone MODEM connections and via the TCP/IP network (NSFnet). IBM PC based software has also been developed and distributed to allow access to and to allow movie loop display of GOES imagery on a scheduled bases. For non IBM PC users or for users with unique applications for the meteorological data, a

McIDAS to NSFnet bridge has been designed and implemented to allow remote users to access McIDAS raw data and information files via the TCP/IP network using the File Transfer Protocol (FTP).

Experiment Results

Successful product transfers have routinely occurred, thereby demonstrating the feasibility of such real time distribution methods. Purdue has recently examined AVHRR data that was obtained via the TCP/IP link during the experiment. There is a desire for these data to be made available routinely. For users with the required IBM PC hardware, successful GOES images were acquired and displayed. Presently, there generally appears to be little operational need of such products by earth science users. However, we believe that since potential users are only beginning to have access to the McIDAS products, the potential for their applications have not been fully examined. If support is found for continuing this public distribution of McIDAS products following the Telescience program (in particular we hope to make AVHRR data available to Purdue routinely), more users may recognize their utility and thus a demand may surface.

Section 4

Infrastructure Experiments

4.1 Communications Architecture (Univ. of Michigan)

The objective of this effort was to determine the requirements for communications between the remote and local sites.

Experiment Description and Summary

We investigated the ability to present a view of the remote coaching workstation on the local experts workstation.

Issue Investigated

What kind of communications architecture is required to communicate between the local and the remote site.

Experiment Hypothesis

An expert could be presented with a view of the state of the system at the remote site. It could be made to look as though he were watching the operation of the technician as he interacts with the expert system at the remote site, giving him additional guidance as needed.

Method of Investigation

Existing communications protocols were investigated to determine if any were applicable to the remote coaching. If not a new communications protocol was defined to satisfy the requirements of this project.

Experiment Results

Several different serial communications protocols were investigated, such as KERMIT, XMODEM, YMODEM and ZMODEM. In each case, it was found they were all designed for file transfer from one computer to another. The remote coaching application is quite different. Not only is the ability to transfer files a requirement, but also the expert and the technician should be able to enter into a dialog while the expert system at the remote site is running. The expert should be provided with a window on his workstation showing the interaction between the technician and the expert system.

Our solution to this problem was to divide the software at each site into four separate programs.

At the local site, where the expert is located, there are four programs running concurrently. The first program is a communications program used to direct information to and from the remaining three programs, through the serial port which is connected to the modem. The connection between these programs, is via TCP/IP based socket calls. This allows communication to a multitude of computers other than the one where the serial port is physically located. The second program receives text from the communications program and displays it in a window on the monitor screen. The third program, prog3, receives text from

the communications program, and displays it in a window on the monitor screen. The fourth program receives text input from the technician and sends it to the communications program to be routed to a text window on the remote technicians workstation.

At the remote site, where the technician is located, there are four programs running concurrently. The first is the communications program which is used to direct information to and from the remaining three programs through the serial port which is connected to the modem. The second program receives text from the communications program and displays it in a window on the monitor screen. The third program receives text from the communications program and displays it in a window on the monitor screen. The fourth program receives input from the operator and sends it to the communications program to be routed to a text window on the local experts workstation.

Thus, there are several windows located at the experts workstation and several located at the technicians workstation. Text which is entered at the technicians talk-local window is displayed at the experts talk-remote window. Thus, text entered in one window of the technicians workstation is routed different windows at the experts workstation. To make all this work we had to define and implement our own communications protocol. The protocol enabled variable length packets of character information to be transmitted to and from a processes running on the remote portable computer back to a specific processes running on the experts host computer.

4.2 Portable Workstations (Univ. of Michigan)

This effort was to determine the feasibility of using a portable workstation at the remote site.

Experiment Description and Summary Conclusion

The required workstation environment for supporting remote coaching was investigated. We did this by reviewing the different manufacturers of portable computers which could satisfy the following requirements.

- Light weight and easy for one person to carry.
- A large variety of expansion boards, such as vision boards, should be available at a modest cost.
- A large amount of software, such as compilers, should be available.

Issue Investigated

What is the possibility of using a portable computer for the remote workstation?

Experiment Hypothesis

A portable computer could be used as the host computer at the remote site for remote coaching. This computer would have sufficient memory and expansion boards which are required to support all of the features of a remote coaching system.

Method of Investigation

A computer system was selected for use by the technician at the remote site. This system was evaluated in terms of the requirements for remote coaching.

Experiment Results

The Compaq Portable III personal computer was selected as the host computer for the technician at the remote site. This particular computer is an IBM AT clone. We found the Compaq to be completely adequate in most regards. The expansion boards for vision and serial I/O were readily available and performed as expected. However, we found two limitations which could be easily resolved.

The first limitation dealt with memory, both RAM and disk. We found that the IBM PC systems were originally designed with the notion that no user programs would ever require more than 640K of RAM. Though the Compaq was equipped with 640K of RAM, we felt that more memory would be needed to support extensions to the expert system. Hence, the use of extended memory was explored. It was found that it is possible to store data in this type of memory, however, all program code must reside within the 640K barrier. Further, a program that wishes to store data in the extended memory space must be specifically written to do so. Since the CLIPS program was obtained from NASA, we could not take advantage of any extended memory.

The 640K RAM limitation was a severe restriction on the implementation of the remote coaching system. The CLIPS expert system program requires 200K to load before any rules are read into the system. The remote coaching rules which were developed for maintenance of the Fabry-Perot interferometer required another 500K of additional memory to load. Thus, as the system currently is designed it is impossible to load the entire remote coaching system. Only smaller pieces of the system can be loaded at a time.

Another memory limitation is the number of video pictures which could be stored. Each picture requires about 250K bytes for storage. Thus, a 20 Mega byte disk can only hold about 80 pictures. A fully operational remote coaching system would require many more pictures.

The second limitation has to do with the DOS operating system used by the IBM PC type computers. The ability of the remote technician to continue working with the expert system while communicating with the real expert implies multitasking, one process for the expert system and the other process for communications. Not only is DOS not a multitasking operating system, it is not reentrant, which makes it difficult to develop multitasking software which uses the system calls provided by DOS. Such system calls are used by all software developed for the PC for such things as terminal I/O. Our temporary solution to this problem was to use a multitasking software package called DESQView. This software runs on top of DOS and performs task switching between system calls by monitoring all system calls made by the users program. This method works fairly well for most programs provided they do not directly access any system memory, such as the monitor display memory or define interrupt routines for such things as serial port drivers which might be swapped out to disk.

Another indirect problem with using DESQView was the additional memory lost for running DESQView, which reduced the amount of memory available to run the remote coaching software. However, the DESQView system did allow us to get a portion of the remote coaching system operational, with communications to the local site to the expert.

Both of these problems could be addressed fairly easily. The memory problem with storing video images on a hard disk could be solved by simply using a video disk. Versions are available, which can easily store thousands of pictures. The problem with the 640K RAM limitation and the multitasking requirements can be solved by using a more powerful operating system such as a UNIX clone. The popular XENIX operating system would be a natural choice. With this system, the Compaq could support up to sixteen megabytes of RAM. This should be plenty of memory for even the most demanding remote coaching system.

4.3 Use of Networks To Enhance Programmatic (Smithsonian Astrophysical Observatory)

Issue Investigated

Although it was not part of our investigation plan, we have found that the network connections between the various participants in the TTPP has greatly enhanced our overall productivity for the program. This is not something that can be easily quantified. However, from our daily experiences with using the network we know that it provides a significant improvement in the day-to-day activities performed separate from activities such as teleoperations or teleanalysis. Specifically:

Electronic mail (correspondence):

E-mail does not present a heavy network load and the network throughputs are satisfactory in both bandwidth and latency for e-mail. E-mail is one of the most useful aspects for making timely contact with other people. E-mail provides the details and self-documentation that has been lost with phone conversations. (Prior to the use of phones, history and documentation was preserved via letter mail records.) Also, unlike a telephone where either a person gets interrupted by the phone or may not be present to answer the phone, e-mail provides a recorded message which can either be acted upon immediately or deferred or is received whether or not the person is present.

Another major aspect of e-mail is the speed which provides nearly instant delivery and thereby speeds up the completion of activities. Many times in dealing with an issue several messages can be transmitted back and forth with an hour, avoiding playing "phone tag", providing written explicit information and a complete record of the "discussion".

Additionally, with e-mail one has electronic copies of the information which can be further distributed, edited or combined with other data, rather than having to re-type the information.

An excellent example of how this was used was in preparation of the area quarterly reports. In this case, each PI wrote his own quarterly. In addition to submitting them to the project office, a copy was sent to the area coordinator who could then use the information contained in the individual PI reports to create an area summary report. As part of the summary, the area coordinator performed a survey of the PIs asking for information on hardware, operating systems, software, network usage, etc. This poll was conducted over the network, results received as e-mail which could then be easily and in a timely fashion collated to produce the area summary report each quarter. Depending on verbal responses or delivery of "US snail" would have made the task much more difficult and time consuming.

Electronic mail (information distribution):

E-mail has been particularly helpful in providing for information distribution. There are several distinctly different categories of information being distributed for a number of differing reasons:

- Meeting agendas, notices, announcements and other forms of general and time critical data which are broadcast with no response expected.
- Documents that are being distributed for review and comment. These could be for document preparation purposes or could be for formal review such as that of proposals. Responses are generally expected or required.
- General distribution of completed documents. This is an area where caution needs to be taken, that is, both from the standpoint of load on the network and filling of disk space at the recipients end. In general, we would recommend only distributing of final documents upon the specific request of the end users rather than the broadcasting approach.

Conclusions With Regard to Effects of E-mail

We feel that e-mail has enhanced dramatically the effectiveness of keeping the program participants informed, of distributing and obtaining information in a timely fashion and in exchanging of files either texts, data or software where an electronic copy rather than a hardcopy is the preferred format.

4.4 Measurement of Network Performance (Smithsonian Astrophysical Observatory)

Issue to Investigate

As we were beginning to use the Internet link between Cambridge and Tucson, we felt that it would be useful to have a measure of the network performance between the two sites, especially since we were anticipating that the performance would be improving over the time of the TTPP. In particular, we expected that the two sites would see an improvement due to completion of the NSF supercomputer interconnections (both SAO through Harvard and University of Arizona are part of the JVNC), connection of University of Arizona to NSN (SAO already has several 56k links to GSFC) and the general improvement in the Internet with the change/over to MERIT in July of 1988.

Experiment Description

We have been monitoring the network performance between Cambridge and Tucson by running of a "ping" program. The "ping" program consists of sending out 10 packets of 512 bytes of data every half hour, that is, 332 packets per week. We then recorded the number of packets that returned and the round trip time for the packet, similar to what one might expect for the response to a key stroke in telnet or for a instrument command.

Experimental Results

We now have data spanning nine months and have seen a significant improvement in both the success rate and in the ping times beginning in the week of 12 June and continuing through July with another improvement beginning in August. The ping data can be divided naturally into four time periods of the first ten weeks of the year for which we have ping data (24 Jan to 28 Mar); the next ten weeks (29 Mar to 11 June), a third group of eight weeks (12 June to 6 August) and the last group of seven weeks (7 August to 26 September). The mean number of times that 100% of the packets pinged was 63% in the first period, 39% in the second, 73% in the third and 75% in the fourth period. The packet loss in the second period was dominated by three consecutive weeks when the network was almost dead with completions of only 2%, 3% and 7% for those weeks.

Although the packet completion rate was not substantially better than the earlier part of the year, the ping times have improved dramatically. The mean time for 10% of the packets to make the round trip (analogous to a key stroke echo) was 860 msec, 780 msec, 683 msec and 416 msec for the four time periods with the mean for the last three weeks of 28 Aug to 17 Sep being an amazing 376 msec. The most significant improvement was for the mean time for 90% of the packets to ping. The values were 2.60 sec, 2.91 sec, 1.074 and 0.874 sec. Thus there appears to have been a substantial improvement in performance since 12 June that has continued.

Over the past year we have also been watching the transfer rate over the network between our two sites. As one might expect, a single user cannot in general realize the full network bandwidth. In tests involving files of about 1/2 Megabyte, the rates are typically on the order of 1.5 kbytes/sec = 12 kbits/sec and the rates have been about the same throughout the year. This is comparable to what one can expect from the best dial-up modems.

We have been waiting for nearly the entire period of the TTPP for installation of the NSI 56k link to the University of Arizona, (SAO already has a 56k tail circuit from GSFC). The word we have now is that this will not be in place until at least mid-October, which is too late to be of any use with regard to our TTPP activities.

Recommendations

Our experience with the networks is that they are very satisfactory for use for e-mail, for some commanding, remote logins and other cases where one is seeking status information. However, they are very inadequate for use to quickly transfer large data sets, such as, interactive image processing, transfer of very large data files like several mega bytes of software. Network performance has to improve before this can be depended upon for routine use.

Both the ping tests and the transfer rate tests have been run using the Internet with no ability to influence the routing. We know that high bandwidth circuits exist between our sites, for example, T1 between SAO and JVNC and, very soon, 56k between JVNC and University of Arizona and alternately, 56k between SAO and the NSN backbone and, soon to be in place, 56k between the NSN backbone and University of Arizona. However, beyond our local gateway, we cannot control how the packets are routed. It would of course be nice to be able to force the routing to use the higher bandwidth circuits and thus improve the throughput between the sites where this is known.

4.5 Telescience Workstation Environment (RIACS)

Conducted at RIACS, this activity consisted of selecting, implementing, distributing, and evaluating the effectiveness and usefulness of a package of software and user interface extensions designed for Sun workstations to enhance one's ability of do productive work in a variety of situations.

Experiment Description and Summary Conclusion

A number of university researchers have small groups with only a couple of workstations. In such an environment it is hard to justify employing the amount of computer systems staff needed to maintain and upgrade the software environment. At such sites, therefore, it is of great value to have pre-packaged software that is readily installed. A second advantage is that, even at sites with substantial systems staff, there is considerable cost savings achieved by not having to search out and obtain the various software packages developed in the research community. Such a service could be of great value to the research community if provided in a central way either by OSSA or through TMIS.

In the present activity the workstation environment required to support a wide variety of interactions between remote users was evaluated. We found that individual productivity might be enhanced in telescience situations by having a package of software available which provided a broad range of functions. In the first phase of this activity we integrated a set of public domain and other free software into an easily installed software package and then made it available by both tape and network file transfer to those in the community using Sun workstations. Additional software developed by the TTPP community and others were then integrated into the package and the upgrade made available.

Approximately half-way through the year, the effort was initiated to assess the acceptability and usefulness of this package called "TeleWEEn" to its users. A carefully formulated survey was designed and distributed to the 15 institutional participants of the TTPP. This second phase of this activity constituted a human-factors oriented assessment of the effectiveness of the workstation environment.

A total of only four survey responses were received by the conclusion of the TTPP. This was probably because most recipients had not had sufficient computer memory, time or manpower to install TeleWEEn. While limited in number, these responses were useful in planning for possible follow-on efforts. In addition, TeleWEEn received positive responses to the following questions. Did TeleWEEn have any direct impact on (1) interaction among TTPP participants, (2) speed of problem solving, (3) accuracy of problem solving, (4) more efficient use of time, (5) positive influence on the effective use of overall labor time, (6) positive influence on overall workload, and (7) positive influence on programming development time? Further survey findings are presented below.

As part of this activity a conceptual validation model was also developed which incorporates both measured and estimated system performance parameters (Haines, 1989).

Issue Investigated

What is the feasibility and usefulness of providing the scientific user community with a pre-packaged set of software that is easily installed and provides the opportunity for upgrade through community-based software developments.

Experiment Hypothesis

It was hypothesized that scientific productivity could be increased in a cost effective manner by having a single organization act as an integration site for a standard workstation environment.

Method of Investigation

A full-time research associate (Mike Slocum) obtained software from various sources. Some of the software was part of the standard UNIX release tape, some was obtained from various public domain software, some was obtained through negotiation with software developers interested in having the TTPP act as a test environment, and some was developed in the TTPP community itself. A list of the specific software included in TeleWEn is given in Table 1.

Table 1
Software Included in TeleWEn

1. Rand MH electronic mail handling package
2. GNU Emacs extendable text editing system
3. TeX document processing system (including public domain fonts)
4. LaTeX document processing system
5. Bibtex (a (La)TeX bibliography processor)
6. Slitex (a LaTeX viewgraph generator)
7. DVI to Postscript filter (to convert (La)TeX output to that suitable for a Postscript printer)
8. TeX utilities
9. Utah graphics tool kit (for applications development)
10. Calen, Wisconsin calendar and appointment scheduling program
11. A symbolic debugger (accompanies GnuEmacs)
12. Kermit (machine to machine file transfer program)
13. Macintosh communication programs (macput and macget)
14. Frame Maker (demo version)
15. Netup (network/host monitoring program provided by RIACS)
16. metafont (not installed as an executable)
17. rdist (remote file distribution service)
18. s21latex (scribe to LaTeX translator)
19. generic user files (to set up telescience user accounts)
20. Documentation on use and installation of TeleWen (provided by RIACS)

This software was integrated into a package that was easily installed on a stand-alone Sun workstation or file server. Documentation was prepared for both the software and its installation. The software was then made available by either mailing tapes or (preferably) through network file transfer to those in the TPP community wishing it. The environment was distributed to 7 TPP subcontractor universities (Purdue, Maryland, Stanford, SAO, UC Berkeley, Cornell, and Arizona), 2 NASA centers (Goddard, JSC) and 2 other organizations collaborating in this investigation (Olivetti Research Center and Space Telescope Science Institute). The environment was assumed to be installed on top of Sun UNIX 3.4 and contained software providing word processing, text editing(WYSIWYG) and others listed in Table 1. After the initial release, several of the recipients brought other packages to our attention and they were integrated into follow-on releases of the software.

After the TeleWE_n package had been distributed for approximately two months, a RIACS human factors research scientist (Richard Haines) designed and administered a user survey to evaluate TeleWE_n's effectiveness in the TPP. This was done using electronic mail so that each respondent could reply with minimum effort and time. A major objective of this activity was to determine the types of uses they made of workstations and the extent they felt such workstations, running the TeleWE_n package enhanced their productivity.

The survey had several objectives which were divided into the Pre-TeleWE_n receivable period (to assess who the TPP participants had conducted their work before receiving the software package) and the Post-TeleWE_n receivable period to assess its impact. Questions were asked about the major activities their computer(s) was used for and which operating systems have been used prior to receiving TeleWE_n. Then an array of questions were asked about the degree of use of the various TeleWE_n components, specific ways in which TeleWE_n may have helped them in applying specific methods or techniques, in evaluating their project, or otherwise meeting their objectives. Questions were also asked regarding pre- and post-receivable of TeleWE_n in terms of whether there was a judged change in their ability to be productive in such areas as "data plotting/graphics," "document preparation," "electronic mail," "general editing," "network monitoring," and "word processing." The results of this survey are given next.

Experiment Results

A total of four completed surveys and one explanatory letter were received. In addition, ad hoc comments also were made by a TPP workshop participant which is presented below. This small response makes it almost impossible to draw any general conclusions concerning the specific survey objectives. Nevertheless, the comments that were made are valuable and are included below. The four respondents were in the occupational and disciplinary categories shown at the headings of Table 2 which presents the results of the survey.

Table 2
Survey Results

Question	Respondent			
	A	B	C	D
1. Occupational Category	PI	Administrator	Research	System Programmer
2. Prime Discipline	Earth Science	Technology	Astronomy	Computer Science
3. Activities Computer is used for:				
Access to other computers	X	X	X	X
Compiling	X		X	X
Conferencing		X		
Data Analysis	X		X	X
Data Collection	X	X		X
Data Generation	X	X	X	X
Data Viewing	X	X	X	X
Document Preparation	X	X		X
Electronic mail processing	X	X	X	X
Experiment Design			X	
Experiment Development			X	
Experiment Evaluation			X	
General Text Editing		X	X	X
Graphical Editing	X	X		
Hardware Evaluation	X			
Software Design			X	X
Software Development	X		X	X
Software Evaluation		X	X	X
Use 3rd Party Software	X	X		X
Use Vendor Utilities	X	X		X
Windowing	X	X	X	
Word Processing		X		
Total =	14	14	14	14

Table 2 (continued)

	A	B	C	D
4. List all operating systems you've used	UNIX VMS	UNIX DOS CPM Apple/MAX HP	UNIX VMS DOS AOS (Data General)	UNIX, VMS
5. When did you receive TeleWEn?	5/88	5/87	3/88	5/88
6. If TeleWEn has not yet been installed, why?	System not yet fully configured.	----	Not enough disk space	Not enough disk space
7. Have others at your site used TeleWEn?	----	Yes	No	No
8. Rate each TeleWEn component re: degree of use:	----	note 1	----	----
9. TeleWEn used to communicate via mail?	----	Yes	No	No
10. Useful in applying specific methods?	----	Very much	not used yet	not used yet
11. Useful in evaluating telescience project?	----	Very much	not used yet	not used yet
12. Useful in meeting telescience objectives?	----	Very much	not used	not used
13. Any direct impact in following areas: Ability to:				
(a) Interact more effectively	----	9	----	----
(b) Solve problems faster	----	9	----	----
(c) Solve problems more accurately	----	N/O	----	----
(d) Use time more efficiently	----	9	----	----

Table 2 (continued)

	A	B	C	D
13. Any direct impact in following areas: (cont'd)				
Ability to:				
(e) Influence overall labor time effectively	-----	9	-----	-----
(f) Influence overall workload	-----	9	-----	-----
(g) Influence programming/development time	-----	N/O	-----	-----
14. List name of software used before receiving TeleWEn and rate it for each listed work element. Then rate TeleWEn's relative capabilities to perform the same work element. There were two responses received this question. See note 2 for the scoring key.				
Work Element	(rating score)	(rating score)		
(a) Data plotting/graphics	None	Internally dev. pkgs.(3)		
(b) Document preparation	emacs/troff (blank)	troff(7)		
(c) Electronic Mail	Mh (blank)	sendmail (7)		
(d) General Editing	emacs (blank)	vi (8)		
(e) Graphical Editing	None (blank)	None		
(f) Network Monitoring	None (blank)	None		
(g) Word Processing	emacs (blank)	None		

Note 1. Each TeleWEn component was rated in terms of how much each had been used to date. Only one respondent (No. 2) had used TeleWEn as follows:

<u>TeleWEn Component</u>	<u>Rating</u>
Calen (calendar program)	very rarely
Frame Maker (graphical editing)	very often
GnuEmacs (editing)	very often
GnuEmacs (mail)	very often
Netup (network monitoring)	moderately often
Rand mail handling service	very often

Note 2. N/O = no opinion, 1 = very low ability to support productivity; 9 = very high ability to support productivity, 0 = no impact, -9 = very negative impact.

15. List specific components of the concept "cost effectiveness".

Respondent 3 answered:

"Providing a set of 'share ware' packages for telescience participants to have access to in itself is cost effective. At a glance, the software looks to be more project management and document production, and not very useful for developing telescience types of experiments."

Respondent 4 answered:

"This means two things: improved scientific productivity at the same or lower cost, where cost is both time and money; and, effectively using a tool that has already amortized its development cost."

16. Describe how has TeleWEn supported your specific discipline activities?

Respondent 2 answered: "Telecommunication"

Respondent 4 answered: "Perhaps if you ask me in 6 months I can give you some real facts!"

17. Describe any and all problems which you have experienced so far in installing or using TeleWEn.

Respondent 4 answered:

"I think the documentation and installation can be improved. Most users probably want specific tools, so the focus should be on partial installations."

18. What else would you like to see included in future releases of TeleWEn?

Respondent 3 answered:

"Suggested data compression algorithms, another pc-to-mainframe communication program (other than Kermit) that takes advantage of the higher speed asynchronous modems."

19. Any other comments?

Respondent 3 answered:

"Response to our request for the TeleWEn was prompt, and I apologize for not being as prompt in experimenting with some of the software. Our testbed program development priorities required that we install other packages first, especially in light of the disk storage limitations."

The following comments and observations refer to the data given in Table 2.

Question 3 dealt with how each user actually used their computer(s) in their daily work. While each of the four respondents checked fourteen activities there was diversity which appeared to be related to their own occupational category in most cases. All four respondents indicated that they used their computer(s) to (a) access other computers, (b) perform data generation, (c) perform data viewing, and (d) perform electronic mail processing. Three respondents indicated that they used their computers to: (a) compile programs, (b) analyze data, (c) collect data, (d) prepare documents, (e) perform general text editing, (f) perform software evaluation, (g) use third party software, and (h) do windowing. In summary, of the 22 categories provided, at least one respondent checked every one suggesting that computers are being used in a very wide variety of ways. Of course it is problematic whether TeleWEn would be considered *the* software package of choice.

Question 4 related to what operating systems the respondent has used. All four had used UNIX, three VMS, two DOS, and one AOS (Data General), Apple- Max and HP.

Question 5 showed that TeleWEn was distributed relatively late in the TTPP activity. Nevertheless, some respondents indicated that the large amount of resident disk space required for TeleWEn prevented them from implementing it as early (or at all) as they would have liked to.

In addition to the four returned surveys, one TTPP workshop participant said that he felt that TeleWEn did not possess enough "value added" to justify the time and expense (memory) required to install it. An unanswered question is whether this individual would use TeleWEn if it was readily available to him.

Discussions and Conclusions

Without a sufficient response rate no survey means very much. This is certainly true here. Nevertheless, the individual comments made by the four respondents are valuable. Also, for a survey such as this one to be valid the respondent must have had sufficient time

actually working with TeleWEn on which to base his or her comments. It is unlikely that this was the case. The one respondent who had TeleWEn long enough to make effective use of it (No. 2) gave it high marks (cf. question 13 especially). This response is very likely valid in light of his/her past range of experience using a variety of computer activities (cf. question 8), operating systems (cf. question 4), and previously used software (cf. question 14). Also of particular note are the statements made by respondent 3 to questions 15 and 18. Namely, that TeleWEn appears to be "...more project management and document production" (oriented) and that TeleWEn could be improved by including "data compression algorithms, another pc - to - mainframe communication program (other than Kermit) to take advantage of higher speed asynchronous modems." These two statement may indicate only the specific needs of one person or they may represent a broadly based perception which should be assessed more fully.

Another general conclusion is that this survey seemed to sample the key areas of concern in a clear and systematic way. It may find use in the future.

In summary, the telescience worker will continue to perform a wide variety of activities that involve computational hardware and software. It would seem beneficial to provide as much standardization as possible to them to help foster collaboration and more efficient communication. TeleWEn represents one step in this direction.

4.6 Rapid Prototyping Testbed Program (RIACS)

RIACS has acted as technical program manager for the overall pilot testbedding activity. This pilot activity was intended to validate the feasibility and utility of engaging the scientific community in a set of rapid prototyping testbeds to investigate critical issues in information systems technologies and requirements in support of science in the Space Station era.

Description and Summary Conclusions

Fifteen universities, under subcontract to USRA, have been conducting a set of rapid-prototyping user-oriented testbeds, aimed at investigating the application of various technologies to critical questions in the information system of the Space Station era. RIACS has acted as technical program manager for this activity, and as such, has been responsible for the overall structure of the activity, information exchange between participants and between the program and related activities (e.g. the Space Station contractors), integration of results and their reporting to NASA, and the evaluation of the overall program.

In summary, it is our opinion that the feasibility and utility of user-oriented rapid-prototyping testbeds, conducted in a coordinated manner and aimed at critical issues of information system design, has been proven beyond question. The breadth and depth of this final report is evidence of the scope of results that are achievable through relatively modest funding increments applied to ongoing scientific research. Furthermore, the effectiveness of using a coordinated program involving users (scientists), developers, and technology researchers working together on such issues has been demonstrated.

A number of results and recommendations concerning the conduct of such a program have emerged and are discussed in detail in Volume II, Section 4, and therefore will not be repeated here.

Issue Investigated

This experiment is best viewed as the programmatic aspect of the overall TTPP, investigating the feasibility and utility of investigating critical issues in information systems for the Space Station era through a user-oriented rapid-prototyping testbed environment. Specific issues investigated were:

- The use of electronic media, including networking and electronic mail, in conducting such a program,
- The utility of incremental funding applied against ongoing scientific research to assess the utility of advanced technologies in the conduct of such research,
- Methods for information flow within and without the program,
- Administrative mechanisms to assure rapid turn around from technical advance to its application to scientific research, and
- Integration of program results and their input to the Space Station requirements process.

Experiment Hypothesis:

It was hypothesized that the combination of talents from the scientific and technology communities could be mobilized through such a program to effectively address critical issues in the design of the information system for the Space Station era.

Method of Investigation:

RIACS acted as technical program manager for the Telescience Testbed Pilot Program (TTPP). The experiment described in this section in fact involved the management and administration of the entire TTPP. Volume II, Section 2 (Program Description) provides a complete description of the program. The method of investigation of the issues of this experiment was to pay careful attention to such issues as we conducted the overall activity.

Experiment Results:

As mentioned above, the experiment was successful in allowing assessment of all the issues described above. The programmatic results described in Volume II, Section 4 give a full accounting of the results of this experiment, and will not be repeated here.

4.7 Telescience Workstation Activities (Stanford)

A portion of the Stanford effort of the Telescience Testbed Pilot Program is directed towards the development of multimedia telescience workstations for a variety of real-time instrument control and monitoring functions. The areas of software portability, ease of interface development, availability of standard software tools are being explored.

Experiment Hypothesis

Computer workstations are becoming an essential part of scientific research because of their ability to display and manipulate large amounts of data. In addition, software packages are being developed that support real-time data display and instrument commanding in a workstation environment. Because future space based scientific experiments will depend on adequate communication with and display of data from remote instruments, computer workstations will be an increasingly important tool. As such, understanding the current state of workstation capabilities and likely direction of evolution will impact how science is accomplished in the near future. In addition, using computer workstations for data analysis as well as data collection and instrument control will make them a general purpose tool. As such is necessary to determine the requirements for realtime data transmission for space operations, and to evaluate the interfaces and resources being planned for science experimenters.

Investigation Method

Several computer workstations are being used in connection with real-time data display, instrument commanding and data analysis. These workstations include DEC VAX workstations running under both the VMS and Ultrix operating systems and Sun 3 and SGI IRIS workstations running under the Unix operating system. A Sun 3/260 workstation was purchased as part of the Stanford effort of the Telescience Testbed program in order to be more compatible with other participants in the program.

The University of Colorado OASIS package has been installed at Stanford University on a GPX VAXstation running the VMS operating system. The examples and interface definitions have been studied in order to understand how to customize the input and displays for specific uses such as the simulation of the Spacelab-2 mission. The intent is to develop command and data display interfaces for instruments that have flown on Spacelab-2 and are being developed for the Shuttle Electrodynamic Tether. The evaluation of the OASIS package is in part to determine how easily it can be interfaced to display science and engineering parameters in a particular application. Another goal is to develop appropriate connections to a payload simulator for commanding purposes.

The Transportable Applications Executive (TAE) software package is also being explored as a means of using computer workstations for spacecraft instrument monitoring and control. A demonstration instrument interface based on TAE has been received from Goddard Space Flight Center and brought up on the Sun 3/260 workstation. This demonstration program is based on the High Resolution Solar Observatory proposed for future Spacelab flights, and provides a simulation of controlling the operation of optical instruments viewing solar features. The demonstration program has been successfully

installed and tested. The complete TAE+ package has been received and has been installed on the Sun 3/260 workstation. It is planned to interface the program to a payload simulator running on another computer in order to provide a more realistic simulation of instrument commanding. In the near future, it is intended to interface the TAE workstation with a computer running an identical payload simulator at JSC.

The Human Information Processing Laboratory's Image Processing System (HIPS) software has been installed on both the Sun workstation and the VAXstation II running the Ultrix operating system. The HIPS software package was developed by the Psychology Department of New York University for displaying and processing images used in perception studies. The software set consists of image transformation tools in the form of standard Unix filters. Routines exist for simple image transformation, filtering, convolution, Fourier and other transform processing, edge detection, line drawing manipulation, noise generation and image statistics computation. The HIPS data format allows image data to be easily moved between various workstations and then to be processed locally. The porting of the software included coding of a display program that would take the standard HIPS image files and output the image on the workstation display under an X windows format. The ability to select a variety of color maps has been included in the display in addition to the ability to tile multiple image windows. Data sets that have been displayed with the HIPS software include digitized pictures, sea ice images from the Seasat satellite, images of auroral ultraviolet emissions from the Dynamics Explorer satellite, and frequency-time spectrograms of Fourier transformed digitized analog signals from the Spacelab- 2 mission.

As part of the effort to develop capabilities to provide coordinated data analysis using extended computer networks, a demonstration program providing a simultaneous display of three-dimensional graphics has been developed on an IRIS workstation. This demonstration of synchronized displays over a distributed network has been developed to determine both the effectiveness and complications of linking workstations. The software interconnects two IRIS 2400 workstations to allow coordinated graphics displays over a variety of networks. The display has both a three dimensional object (the earth with propagation paths of very low frequency radio waves) and text windows. The orientation and size of the object can be controlled from either workstation, and the text windows allow communication between the two systems. The software has been tested on the local Stanford ethernet between two workstations and between workstations at the SUNSTAR Laboratory at Stanford and the Data System Technology Laboratory at Goddard Space Flight Center.

Results

Several interesting problems developed in the course of the communications program development. The first is that the floating point representations under the VAX architecture are different than those used the Silicon Graphics and Sun workstations which use the IEEE floating point standard. The difference could be easily solved by converting the numbers at the source workstation, but it complicates the development of a portable code for standard communications. The second problem is that spawned processes interact differently under the Unix operating system than under the VMS operating system. An evaluation of the code is underway to determine if there is a program structure that can minimize the differences in

the code to make the code more portable, but it requires extensive knowledge of the computer systems. In general, making software more portable often imposes restrictions that prevent the most efficient use of the computer system.

The existing software for real-time display of data and control of instruments is very capable, however hardware and operating system dependencies make these software very non-portable. In addition, the learning curve to use existing packages is still very steep, and for simple applications it is still easier to write new software.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix A Program Participants

University Subcontractors

Cornell University

Terry Herter (PI)

Justin Schoenwald

Massachusetts Institute of Technology (MIT)

Charles M. Oman (PI)

Laurence R. Young (PI)

Byron K. Lichtenberg (PI - Payload Systems, Inc.)

James Elliott (PI)

Hale Bradt (PI)

Anthony P. Arrott (Payload Systems, Inc.)

Paul Boutin

Ben Davis

Feroza A. Deen

Mathew George

Barbara J. Glas

Matt E. Hodges

Braden J. McGrath

Sherwood A. Modestino

Vinit Nijhawan (Payload Systems, Inc.)

Robert L. Renshaw

Ilya Shubentsov

Jacqueline A. Stewart

Purdue University

Richard H. Collier (PI)

Gary M. Bass

Larry L. Biehl

David W. Brown

Rick S. Gifford

Chris J. Johannsen

David A. Landgrebe

Victor M. Mendoza-Grado

Lorraine K. Van Meter

Mahesh Veerina

Phil J. Wyss

Rensselaer Polytechnic Institute (RPI)

Richard C. Hahn (PI)

R. H. Doremus
Martin E. Glicksman
Bruce Herbach
Heribert Wiedemeier

Smithsonian Astrophysical Observatory (SAO)

David Koch (PI)

Steve Criswell
Giovanni Fazio
John Geary
Kirk Gilmore
Charles Maxon
Steve Willner
Bill Wyatt

Stanford University

Peter M. Banks (PI)

Rock Bush
Mike J. Wiskerchen

University of Arizona (Astronomy)

Erick T. Young (PI)
George H. Rieke

University of Arizona (Engineering & Mines)

Larry C. Schooley (PI)

Richard Bierenz
Francois Cellier
Reza Fardid
Byron Hack
Jerry Hunter
Alfie Lew
YaDung Pan
Ephraim Raize
Don Schultz
Wendy Walker

University of California, Berkeley (UCB)

Supriya Chakrabarti (PI)

Carl Dobson
Walter Herrick
J. Garrett Jemigan
George Kaplan
Michael L. Lampton
Roger Malina
Will Marchant
Herman Marshall
Jerry Penegor
Dan Quinlan

University of California, Santa Barbara (UCSB)

John E. Estes (PI)
Jeff Star (PI)

Mark Friedl
Kenneth McGwire
David Stoms

University of Colorado

Elaine R. Hansen (PI)

John R. Cowley
Randal Davis
Phil Evans
Jack A. Faber
Chris Grasso
Eric Hillis
Alain J. Jouchoux
David Judd
Pieter Kallemeyn
Margi Klemp
Fred Lacy
George Ludwig
Jane Pavlich
Pat Sirr
Gail Tate
John Wisneski
Kwok Wong

University of Maryland

Michael F. A'Hearn (PI)

Roger A. Bell
Jason Meyer
David M. Zipoy

University of Michigan

Richard A. Volz (PI)

George Carignan
Lynn Conway
Mike Walker

University of Rhode Island

James G. Kowalski (PI)
Peter C. Cornillon

University of Wisconsin

Verner E. Suomi (PI)

Ralph Dedecker
William L. Smith

USRA Program Management

USRA Headquarters

David Holdridge

Deborah Cook
Beverly Johnson

Research Institute for Advanced Computer Science (RIACS)

Barry M. Leiner

Robert L. Brown
Dee Doyle
Lorraine L. Fisher
Maria L. Gallagher
Michael A. Slocum

Affiliated Organizations

NASA/Ames Research Center (ARC)

Vicki L. Johnson (RIACS)
Richard F. Haines (RIACS)
Frank King
Daryl Rasmussen
John Stauffer (SIRTF Project Office)

NASA/Johnson Space Center (JSC)

Floyd Bednarz

NASA/Kennedy Space Center (KSC)

Ken Anthony (Lockheed)

Richard J. Brown

Carol Carlson (EG&G)

Luis Delgado

Richard L. Fiser

Matthew S. Galloway

Douglas J. Gruendel

Robert Marzig (Grumman)

William R. Munsey

Marc G. Timm

David A. Tipton

Deborah S. Vordermark (Bionetics)

NASA/Lewis Research Center (LeRC)

M. J. Bonner

J. C. Johnston

Thomas Glasgow

California Institute of Technology (Cal Tech)

T. N. Gautier, III

University of Rochester

William J. Forrest

THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix B

Glossary

AAS	American Astronomical Society
AGC	Automatic Gain Control
AIPS	Astronomical Image Processing System
ALOT	Arc Laser Optical Technology
Andrew	Multimedia mail system; basis of Carnegie-Mellon EXPRES system
ARC	Ames Research Center
ARPANET	Wide area data network supported by DARPA
AT	Astrometric Telescope
ATF	Astrometric Telescope Facility
Athena	MIT student network
AVHRR	Advanced Very High Resolution Radiometer; on the nimbus series of satellites. Operated by NOAA
B&W	Black and White Display
BARRNET	Bay Area Regional Research Network
BAUD	A unit of signaling speed; refers to the number of times the state or condition of the line changes per second
BCE	Bench Checkout Equipment
BDCF	Baseline Data Collection Facility (at KSC)
CAS	Canadian Astronomical Society
CCD	Charge Couple Device; a technology for converting images into electrical signals
CCSDS	Consultative Committee for Space Data Systems
CDP	Command, Data, and Power interface unit; part of the EUVE instrument
CLIPS	A programming language for expert systems..
CODEC	Coder/Decoder
CSDF	Commercial Space Development Facility
CUI	Common User Interface
DARPA	Defense Advanced Research Projects Agency
DEC	Digital Equipment Corporation
DMIL	Direct Manipulation Interface Language
DOC	Discipline Operations Center
DSP	Digital Signal Processing
EPS	Experiment Payload Specialist
EUV	Extreme Ultraviolet
EUVE	Extreme UltraViolet Explorer

EXPRES	Experimental Program in Electronic Submission
FUV	Far Ultraviolet
FRICC	Federal Research Internet Coordinating Committee
GOES	Geostationary Operational Environmental Satellite
GPX	Graphics Processor Workstation for microVAX II
GSFC	Goddard Space Flight Center
HCIG	Human-Computer Interface Guide
HIPS	Human Information Processing Laboratory's Image Processing
HRPT	High Resolution Picture Transmission
HUP	Human Use Protocols
IBM	International Business Machines
ICD	Interface Control Document
IDL	Interactive Data Language
IGBP	International Geosphere Biosphere Program
IL	Intermediate Language
IMS	Instrument Management Services
Ingres	A database
IOMS	Instrument OMS
IPAC	Infrared Processing and Analysis Center at Caltech
IRAF	Image Reduction and Analysis Facility
IRAS	Infrared Astronomy Satellite
JPL	Jet Propulsion Laboratory
JSC	Johnson Space Center
JVNCC	John Van Neuman Computing Center
KSC	Kennedy Space Center
Kermit	A file transfer program
Kiwi	A "flightless bird" consisting of prototype EUVE electronics
LAN	Local Area Network
LASP	Laboratory for Atmospheric and Space Physics
LCC	Local Controlling Computer
LIB\$TPARSE	VAX/VMS library routine that provides a table driven parser. Used for the initial version of the CSTOL parser for OASIS
LSTB	Life Sciences Testbed
Magic/L	Interactive programming language developed by Loki Engineering, Inc
McIDAS	Man Computer Interactive Data Access System
MIT	Massachusetts Institute of Technology
MMSL	Microgravity Materials Science Laboratory
MMT	Multiple Mirror Telescope on Mt. Hopkins, AZ
MSFC	Marshall Space Flight Center

NASA	National Aeronautics and Space Administration
NASA SELECT	NASA operated TV channel which carries NASA related events
NASCOM	NASA Communications- mission critical
NFS	Network File System
NICOLAS	The inter-network gateway at Goddard Space Flight Center
NOAA	National Oceanic and Atmospheric Administration
NOAA-G	Name of the NOAA polar orbiting satellites
NRAO	National Radio Astronomy Observatory
NSE	Network Software Environment
NSF	National Science Foundation
NSFnet	Computer network supported by NSF
NSI	NASA Science Internet
NSN	NASA Science Network; TCP/IP part of NSI
NTSC	Standard video signal format
OASIS	Operations and Science Instrument Support
OMS	Space Station Operation Management System
OMS/PMS	Operations Management/Platform Management System
OMSS	Operation Management System Services
OSSA	Office of Space Science and Applications
PI	Principal Investigator
PSI	Payload Systems, Inc.
RA	Research Assistant
RCC	Remote Commanding Computer
RFH	Remote Fluid Handling
RGB	Red, Green, Blue Video Display
RIACS	Research Institute for Advanced Computer Science
ROM	Read Only Memory
RS-232	Standard for serial data transmission
SAIS	Science and Applications Information Systems
SAO	Smithsonian Astrophysical Observatory
SCS	Software Control System
SIMBAD	A cross-referenced database of 700,000 stellar and 100,000 non-stellar objects
SESAC	Space and Earth Sciences Advisory Committee
SFDU	Standard Formatted Data Unit
SME	Solar Mesosphere Explorer satellite
SOP	Science Operations Subgroup
SPAN	Space Physics Analysis Network

SPIE	Society of Photo-Instrumentation Engineers
SS	Space Station
SSE	Software Support Environment
SSIS	Space Station Information Systems
SSL	Space Sciences Laboratory at UC, Berkeley
SSP	Space Station Program
STScI	Space Telescope Science Institute
TAE	Transportable Applications Executive
TATS	Thaw Atmospheric Telescope Simulation
TCP/IP	Transmission Control Protocol/Internet Protocol
TDRSS	Tracking and Data Relay Satellite System
TeleWEn	Telescience Workstation Environment
Terracom	A company name
TFSUSS	Task Force on Scientific Uses of Space Station
TIF	Telescope Interface Unit
TIGS	Testbed at LASP
TMIS	Technical Management Information System
TPPP	Telescience Testbed Pilot Program
UC	University of California
UCB	University of California, Berkeley
UCSB	University of California, Santa Barbara
UIL	User Interface Language
Unify	A database program
UofA	University of Arizona
USE	User Support Environment
USRA	Universities Space Research Association
UW	University of Wisconsin
VISTA	another image processing system
WAN	Wide Area Network
ZOE	Zone of exclusion

Appendix C Bibliography

Bennett, Elliot, *HOLOP: A Case Study of Advanced User-interface Design for Interactive Control of Space Experiments*, DFVLR IB 333 - 88/5, 24 pp., DFVLR, Institute for Space Simulation, Cologne, West Germany, June 1988.

This paper details advanced concepts in user-interface design implemented in the computer program HOLOP Ops. HOLOP Ops was designed to provide a simple, easy, and fast user-interface for remote, interactive control the HOLOP facility aboard the D2 Spacelab mission. Such a user-interface is achieved by implementing full graphics capabilities (including pictures, icons, graphs, and mouse/cursor control) as well as full text displays and control in a transparent, integrated environment for experiment observation and control. The advantages and capabilities of this program's user-interface are described and analyzed for their ability to enhance space based science productivity in the Space Station era.

Bienz, Richard A. and Larry C. Schooley, "A Survey of Computer Networks," Technical Report TSL-001/87, 85 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, February 1987.

A summary of existing wide-area computer networks and their attributes and evaluation of their possible use in the University of Arizona TTPP testbeds.

Bienz, Richard A., and Jerry J. Hunter, "Communication Software Design for Telescience Demonstrations," Technical Report TSL-019/88, Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, December 1988.

Cellier, Francois, *Teleoperation of the Thaw Telescope at the Allegheny Observatory: A Case Study*, Technical Report TSL-004/87, 56 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, December 1987.

Primary purpose of this document is to define the intermediary language to be used for computer-to-computer communication between the local controlling computer at Allegheny Observatory and the remote commanding computer which will be located at the University of Arizona. Overview of plans leading to teleoperation the Thaw Telescope at Allegheny Observatory is also discussed as well as control loops, sensors, safeguard error messages.

Chakrabarti, Supriya, Carl A. Dobson, and J. G. Jernigan, "Telescience at the U. C. Berkeley Space Science Laboratory," *Bulletin of the American Astronomical Society*, vol. 19, p. 744, Space Sciences Laboratory, U.C. Berkeley, June 1987.

The University of California, Berkeley is a member of a University consortium developing methodologies for remote design, development and operation of space instrumentations, collectively termed *telescience*. We will discuss our efforts in extending an existing local software control system to allow the development and sharing of software between remote sites. We are developing a methodology for the remote operation of instrumentations utilizing networks such as the ARPANET. These techniques have already been demonstrated over a local Ethernet. These two areas of investigations address the teledesign and teleoperation components of telescience.

Chakrabarti, Supriya, Carl A. Dobson, George C. Kaplan, Herman L. Marshall, Michael Lampton, Roger F. Malina, Stuart Bowyer, and Jeff Star, "Astronomical Data Analysis from Remote Sites," *Astronomy from Large Databases, Scientific Objectives and Methodological Approaches, ESA Conference and Workshop Proceedings*, vol. 28, p. 295, Space Sciences Laboratory, UC Berkeley & Remote Sensing Research Unit, UC Santa Barbara (J. Star), Berkeley, CA, January (1988c). Also available in Space Astrophysics Group Contribution Number 329.

Given the progress in the communication technology, it is expected that during the space station era the mode of instrument operation and data analysis will be dramatically different. A consortium of several universities and NASA centers are evaluating various aspects of design and operation of scientific instruments and data analysis over various computer networks from a remote site. Such a scheme has officially been termed telescience. We will report on the development of methodologies for teledesign, teleoperation and teleanalysis and the verification of these concepts using the Extreme Ultraviolet Explorer (EUVE), a satellite payload scheduled for launch in 1991. The EUVE telescopes will be operated remotely from the EUVE Science Operation Center (SOC) located at the University of California, Berkeley. Guest observers will remotely access the EUVE spectrometer data base located at the SOC. Distributed data processing is an integral part telescience. We will describe our experience with the Browse system; currently being developed at the University of California at Santa Barbara through a grant from NASA for remote sensing applications. We will discuss the suitability for its adoption for astronomy applications. Browse allows the examination of a subset of the data to determine if the data set merits further investigation. The examination can be as simple as looking for a specific data element based on its location, date of observation, quality indicator, spectral coverage etc. It also allows the viewing of data in various modes depending upon the available resources at the user's end (e.g., graphics terminal vs. dumb terminal), level of data compression applied, required display format etc. and its transmission over a network to a local graphics display station.

Chakrabarti, Supriya, William T. Marchant, George C. Kaplan, Carl A. Dobson, J. Garrett Jernigan, Michael L. Lampton, and Roger L. Malina, "Telescience at the University of California, Berkeley," *Acta Astronautica*, in press, (1988a). Also available in Space Astrophysics Group Contribution Number 356.

The University of California at Berkeley (UCB) is a member of a university consortium involved in telescience testbed activities under the sponsorship of NASA. Our Telescience Testbed Project consists of three experiments using flight hardware being developed for the Extreme Ultraviolet Explorer project at UCB's Space Sciences Laboratory. The first one is a teleoperation experiment investigating remote instrument control using a computer network such as the Internet. The second experiment is an effort to develop a system for operation of a network of remote workstations allowing coordinated software development, evaluation, and use by widely dispersed groups. The final experiment concerns simulation as a method to facilitate the concurrent development of instrument hardware and support software. We describe our progress in these areas.

Chakrabarti, Supriya, William T. Marchant, Carl A. Dobson, George C. Kaplan, Michael L. Lampton, and Roger L. Malina, "Remote Command and Telemetry Handling for a Spaceflight Instrument," *Proceedings of IECON'88: Control and Simulation, Singapore*, v III, pp. 325, Space Sciences Laboratory, UC Berkeley, Berkeley, CA, (1988b). Also available in Space Astrophysics Group Contribution Number 365.

The Space Sciences Laboratory at the University of California, Berkeley, is a member of a university consortium involved in telescience testbed activities under the sponsorship of NASA. As part of our activities, we have developed methodologies for remotely commanding a space-based instrument and receiving telemetered data. Two experiments were conducted to interact remotely with a flight-destined instrument. In the first experiment we sent commands using the Bay Area Regional Research network from a computer at Stanford University to an instrument connected to a workstation located

at the University of California, Berkeley. In the second experiment we used the Internet to conduct the same experiment from the University of Colorado, Boulder. In addition to telemetry, low-rate video images of the instrument were transmitted over the same network to provide visual feedback. Although further testing is necessary, our limited experience indicates that it will be possible to interact with a space-based instrument from an experimenter's desk.

Chakrabarti, Supriya, D. Cotton, M. Lampton, O. Siegmund, R. Link, and G. R. Gladstone, "Remote Sensing of Atmospheric Oxygen from a Sounding Rocket," *Acta Astronautica*, in press, (1988d).

Davis, R., J. Faber, E. Hansen, A. Jouchoix, and G.H. Ludwig, *Telescience Testbed Program Results*, LASP Report 89-1, University of Colorado, Boulder, CO, January 1989.

Forgy, C. L., "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence* 19, 1982, pp. 17-37.

Gallagher, Maria L., *Telescience Testbed Kickoff Meeting Minutes*, RIACS TR 87.25, 26 pp., RIACS/USRA, Moffett Field, CA, September 1987.

The kickoff meeting for the Telescience Testbed Pilot Program was held on July 30-31, 1987 at NASA Ames Research Center. These are the minutes of that meeting.

Gallagher, Maria L., *Telescience Testbed Pilot Program Meeting II Minutes*, RIACS M88.1, RIACS/USRA, Moffett Field, CA, March 1988.

The TPP II meeting was held on March 7-9, 1988 in Boulder, Colorado. These are the minutes of that meeting

Gallagher, Maria L. and Barry M. Leiner, *Telescience Testbed Pilot Program First Quarterly Report*, RIACS TR 87.26, 35 pp., RIACS/USRA, Moffett Field, CA, September 1987.

The Telescience Testbed Pilot Program participants are required to issue reports to NASA Headquarters on a quarterly basis. This is the first quarterly report, covering the period April 28, 1987 through August 31, 1987.

Gallagher, Maria L. and Barry M. Leiner, *Telescience Testbed Pilot Program Second Quarterly Report*, RIACS TR 87.31, 57 pp., RIACS/USRA, Moffett Field, CA, December 1987.

The Telescience Testbed Pilot Program is an innovative activity involving fifteen universities in user-oriented rapid-prototyping testbeds to develop the requirements and technologies appropriate to the information system of the Space Station era. The Telescience Testbed Pilot Program is required to issue progress reports to NASA Headquarters on a quarterly basis. This is the second quarterly report, covering the period September 1, 1987 through November 30, 1987.

Gallagher, Maria L. and Barry M. Leiner, *Telescience Testbed Pilot Program Third Quarterly Report*, RIACS TR 88.8, 77 pp., RIACS/USRA, Moffett Field, CA, March 1988.

The Telescience Testbed Pilot Program is required to issue progress reports to NASA Headquarters on a quarterly basis. This is the third quarterly report, covering the period December 1, 1987 through February 29, 1988.

Gallagher, Maria L. and Barry M. Leiner, *Telescience Testbed Pilot Program Fourth Quarterly Report*, RIACS M88.4, 69 pp., RIACS/USRA, Moffett Field, CA, June 1988.

The Telescience Testbed Pilot Program is required to issue progress reports to NASA Headquarters on a quarterly basis. This is the fourth quarterly report, covering the period March 1, 1988 through August 31, 1988.

Gallagher, Maria L. and Barry M. Leiner, *Telescience Testbed Pilot Program Fifth Quarterly Report*, RIACS M88.5, 76 pp., RIACS/USRA, Moffett Field, CA, September 1988..

The Telescience Testbed Pilot Program is required to issue progress reports to NASA Headquarters on a quarterly basis. This is the fifth quarterly report, covering the period September 1, 1988 through December 31, 1988.

Hack, B., *Man to Machine, Machine to Machine and Machine to Instrument Interfaces for Teleoperation of a Fluid Handling Laboratory*, Technical Report TSL-014/88, Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, June 1988.

The purpose of this thesis is the design and description of the software necessary for teleoperation of a remotely operated fluid handling laboratory. It does not include the implementation of this software. The laboratory for which it is designed is currently being developed at the University of Arizona, and is intended to be a small scale model of the fluid handling laboratory which will be aboard Space Station. The designed software includes a man/machine interface, a machine/machine interface, and a machine/instrument interface. The man/machine interface is graphical in nature, menu driven, and consists of high level commands which are independent of the devices in the laboratory. The machine/machine interface is also device independent. It consists of intermediary commands and maps the commands of the man/machine interface to low level, device dependent, commands and programs of the machine/instrument interface. Although the software is primarily designed for the model fluid handling laboratory, the needs and requirements of the operation of a similar laboratory aboard Space Station have been considered.

Haines, Richard F., *Human Performance Validation Procedures Applicable to Telescience Activities*, RIACS TR (in final review), January 1989.

Johnson, Vicki and John Bosley, "A Shared-World Conceptual Model for Integrating Space Station Life Sciences Telescience Operations," *Proc. 1988 Goddard Conference on Space Applications of Artificial Intelligence*, Goddard Space Flight Center, May 1988.

Jouchoux, A., *Ada Compiler Choice*, LASP Report, University of Colorado, Boulder, Co, January 1988.

Kallemeyn, P.H., B. Knapp, and G.H. Ludwig, *User's Manual - Science Data Base Access Program for the Solar Mesosphere Explorer*, LASP Report 89-3, University of Colorado, Boulder, Co, April 1989.

Kaplan, George C., "EUVE Contributions to Telescience," *EUVE Technical Bulletin*, no. 4, p. 2, Space Sciences Laboratory, UC Berkeley, Berkeley, CA, September 1987.

Brief Overview of UC Berkeley's telescience experiments for the TPP.

Koch, David, Terry Herter, John Stauffer, and Erick Young, *Telescience Applied to the Space Infrared Telescope Facility*, 8 pp., Smithsonian Astrophysical Observatory (Koch); Department of Astronomy, Cornell University (Herter); NASA/Ames Research Center (Stauffer); Steward Observatory, University of Arizona (Young), September 1987.

In the future, the approach to the conduct of scientific space missions will be substantially different from the approach that has been used in the past. A more distributed approach will be taken with the scientists conducting operations and analysis, remotely from their home institutions, making major use of standardized software and compatible hardware. Key to this approach have been the rapid adoption of the use of local and wide area networks, the use of standardized software tools and the plethora of powerful workstations. These developments will be applied to the Space Infrared Telescope Facility project in the space station era. A number of telescience testbed activities are being undertaken to develop experience and to determine the applicability of telescience methods.

Leiner, Barry M., *Telescience Testbed Pilot Program*, RIACS TR 87.12, 42 pp., RIACS/USRA, Moffett Field, CA, May 1987.

The Telescience Testbed Pilot Program is an innovative activity to address a number of critical issues in the conduct of science in the Space Station era. Several scientific experiments using advanced information processing and communications technologies will be carried out and the results evaluated to determine the requirements and their priorities. This will provide quantitative evidence as to the relative importance of different functions in the SSIS and their required performance. Furthermore, it will allow a set of scientific users to gain experience with advanced technologies and their application to science. This report is based on the proposal from USRA to NASA for the establishment of the Telescience Testbed Pilot Program. It describes the motivation for the program, the structure of the effort, and several strawman scientific experiments that constitute the heart of the activity.

Leiner, Barry M. and James R. Weiss, *Telescience Testbedding: An Implementation Approach*, RIACS TR 88.2, 9 pp., RIACS/USRA (Leiner) & NASA/HQ (Weiss), Moffett Field, CA, February 1988.

Telescience is the term used to describe a concept being developed by NASA's Office of Space Science and Applications (OSSA) under the Science and Applications Information System (SAIS) Program. This concept focuses on the development of an ability for all OSSA users to be remotely interactive with all provided information system services for the Space Station era. This concept includes access to services provided by both flight and ground components of the system and emphasizes the accommodation of users from their home institutions. Key to the development of the Telescience capability is an implementation approach called rapid-prototype testbedding. This testbedding is used to validate the concept and test the applicability of emerging technologies and operational methodologies. Testbedding will be used to first determine the feasibility of an idea and the applicability to real science usage. Once a concept is deemed viable, it will be integrated into the operational system for real time support. It is believed that this approach will greatly decrease the expense of implementing the eventual system and will enhance the resultant capabilities of the delivered systems.

Leiner, Barry M., *Telescience Testbed Pilot Program Interim Report*, RIACS TR 88.6, 16 pp., RIACS/USRA, Moffett Field, CA, February 1988.

The Universities Space Research Association (USRA), under sponsorship from the NASA Office of Space Science and Applications, is conducting a Telescience Testbed Pilot Program. Fifteen universities, under subcontract to USRA, are conducting a variety of scientific experiments using advanced technology to determine the requirements and evaluate trade-offs for the information system of the space station era. This report represents an interim set of recommendations based on the experiences of the first six months of the pilot program.

Lew, A. K., *Astrometric Telescope Simulator for the Design and Development of Telescope Teleoperation*, Technical Report TSL-016/88, Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, September 1988.

Lichtenberg, Byron K., "Concepts for Crew Experiment Interaction-Future Space Flights: Workstation Design and Requirements," p. 3, Payload Systems, Inc., June 1988.
Submitted to the International Conference on Environmental Systems to be held July 1988.

Current space lab and space shuttle workstations are inadequate for the next generation of space experimentation. The capability of the current experiment computers is severely limited, the maximum sample rate that can be acquired and processed for on board display is 10 samples per second, and the displays have a maximum of 750 word storage associated with them. Second, the ability to modify experiment operations real time is nonexistent unless it was programmed in approximately 18 months before flight. The appearance of new generations of computers will alleviate these problems,

but acceptance by the space engineering community is still limited. This paper will discuss the concepts and requirements for future workstations and capabilities that should be inherent in the next generation of space craft.

Marchant, Will, Carl A. Dobson, Supriya Chakrabarti, and Roger F. Malina,

"Telescience - Concepts and Contributions to the Extreme Ultraviolet Explorer Mission," *SPIE Proceedings*, vol. 851, p. 173, Astronomy Dept. & Space Sciences Laboratory, UC Berkeley, Berkeley, CA, November 1987. Also available in Space Astrophysics Group Contribution Number 315.

A goal of the telescience concept is to allow scientists to use remotely located instruments as they would in their laboratory. Another goal is to increase reliability and scientific return of these instruments. In this paper we discuss the role of transparent software tools in development, integration, and post-launch environments to achieve hands on access to the instrument. The use of transparent tools helps us to reduce the parallel development of capability and to assure that valuable pre-launch experience is not lost in the operations phase. We also discuss the use of simulation as a rapid prototyping technique. Rapid prototyping provides a cost-effective means of using an iterative approach to instrument design. By allowing inexpensive production of testbeds, scientists can quickly tune the instrument to produce the desired scientific data. Using portions of the Extreme Ultraviolet Explorer (EUVE) system, we examine some of the results of preliminary tests in the use of simulation and transparent tools. Additionally, we discuss our efforts to upgrade our software "EUVE electronics" simulator to emulate a full instrument, and give the pros and cons of the simulation facilities we have developed.

Pan, Ya-Dung, *Teleoperation of Mechanical Manipulators Aboard the U.S. Space Station*, Technical Report TSL-002/87, 74 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, December 1987.

This study presents a new analytical controller design strategy for the teleoperation of mechanical manipulators aboard the U.S. space station. This controller design strategy emphasizes on the stability of a closed-loop control system involving time delay. Simplified dynamic equations of the Stanford arm are considered as the manipulator model. A local linearizing and decoupling control algorithm is applied to linearize and decouple the dynamic equations. Once the linear form of the manipulator is obtained, a model prediction control loop is constructed and implemented as a digital controller to provide the predictive states information, and a particular model reduction method is applied to yield a reduced-order digital controller. This reduced- order digital controller is a highly self-tuned controller which can control the closed-loop system with time delay by following a specified performance.

Pan, Ya-Dung and Alfie K. Lew, *Teleoperations Software for Remote Fluid Handling*, Technical Report TSL-020/88, Electrical and Computing Engineering Department, University of Arizona, Tucson, AZ, December 1988.

Pennisi, Liz, "Computers on Long-Distance Research," *LQP*, p. 8, December 7, 1987.

Magazine article on the use of computers on long-distant research at the University of Arizona.

Raize, Efraim, *Computer Interface for Electrophoresis Apparatus*, Technical Report TSL-011/88, 28 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, May 1988. The author is a visiting scholar at the University of Arizona.

This report summarizes the considerations required for an adequate interface, lists the electronics design and shows the drawings and procedures for operation and maintenance of an Electrophoresis machine in an automated laboratory.

Raize, Efraim, *Syringe Driver Assembly for Automatic Fluid Handling Laboratory*, Technical Report TSL-012/88, 17 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, May 1988. The author is a visiting scholar at the University of Arizona.

This report describes the design and implementation of a driver assembly for an automated fluid handling laboratory.

Rasmussen, Daryl, Arshad Midan, and John Bosley, "Telescience Testbedding for Life Science Mission on the Space Station," *AIAA Aerospace Science Meeting*, Reno, NV, January 1988.

Rasmussen, Daryl, Vicki Johnson, and Arshad Mian, "Telescience Concept for Habitat Monitoring and Control," *18th Intersociety Conference on Environmental Systems*, San Francisco, CA, July 1988.

Schmerling, Erwin, "The Interaction of Users with Instruments and Databases in Space," *Information Systems Newsletter*, pp. 12-18, NASA Headquarters, January 1988.

This article discusses the Telescience Testbed Pilot Program's objectives, planned contributions and defines the term Telescience.

Schmerling, Erwin, "Telescience in the Space Station Era," EASCON, pp. 1-6, NASA Headquarters, September 1988.

After over a quarter of a century of experience in space, and the rapid development of Information Systems capabilities, there is a naturally growing demand for the development of systems where, to an increasing extent, participants can access their fellow scientists and the appropriate NASA service before flight, during flight and after flight, preferably from their home institutions and through the same equipment. This concept has become known as Telescience, and sporadic examples of its implementation may be found in earlier programs.

Schooley, Larry C. and Francois Cellier, *Telescience Testbed Pilot Program Quarterly Report*, Technical Report TSL-003/87, 16 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, December 1987.

First quarterly report for the University of Arizona's Telescience Testbed Pilot Program.

Schooley, Larry C., Richard A. Bienz, and Francois Cellier, *Basic Research in Telescience Testbed Program Final Report: NASA Grant NAGW-1073*, Technical Report TSL-005/88, Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, January 1988.

Final report for NASA grant NAGW-1073.

Schooley, Larry C., Don G. Schultz, and Francois Cellier, *University of Arizona Presentation, Second Telescience Testbed Pilot Program Meeting*, Technical Report TSL-007/88, 50 pp., Electrical and Computer Engineering, University if Arizona, Tucson, AZ, March 1988.

The set of transparencies presented by the University of Arizona at the second TPP meeting held in Boulder, CO on March 7-9, 1988.

Schooley, Larry C. and Francois Cellier, *Telescience Testbed Pilot Program Quarterly Report For Winter 1987-88*, Technical Report TSL-008/88, 15 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, March 1988.

Second quarterly report for the University of Arizona's Telescience Testbed Pilot Program.

Schooley, Larry C. and Francois Cellier, *Telescience Testbed Pilot Program Quarterly Report for Spring 1988*, Technical Report TSL-013/88, 10 pp., Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, June 1988.

This is the third quarterly report for the astrometric telescope, remote fluid handling, and technology development projects at the University of Arizona. It does not include the UA involvement in the SIRTF project.

Schooley, Larry C. and Francois Cellier, *Telescience Testbed Pilot Program Quarterly Report For Summer 1988*, Technical Report TSL-018/88, Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, September 1988.

Summer 1988 quarterly report for the University of Arizona's Telescience Testbed Pilot Program.

Schooley, Larry C. and Francois Cellier, *Telescience Testbed Pilot Program Final Report*, Technical Report TSL-021/88, Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, December 1988.

Final report for the University of Arizona's Telescience Testbed Pilot Program.

Schooley, Larry C., and Francois E. Cellier, *Teleoperation of a Simulated Astrometric Telescope*, Technical Report TSL-022/88 (videotape), Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, December 1988.

Schooley, Larry C., and Francois E. Cellier, Don G. Schultz, *Teleoperation of a Fluid Handling Laboratory*, Technical Report TSL-023/89 (videotape), Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, January 1989.

Schultz, Don G. and R. Fardid, *An Automated Remote Fluid Handling System*, Technical Report TSL-015/88 (videotape), Systems and Industrial Engineering Department, University of Arizona, Tucson, AZ, July 1988

Secord, Terry, *Life Sciences Facility Control and Telescience Systems*, Technical Report MDCH3658, McDonnell Douglas, Huntington Beach, CA, July 1988.

Starks, Scott, David Elizandro, Barry M. Leiner, and Michael Wiskerchen, "Computer Networks for Remote Laboratories in Physics and Engineering," 1988 Annual Conference of the American Society for Engineering Education, June 1988, (also available as RIACS TR 88.13, 7 pp., April 1988).

As we embark on a new era in engineering education, we must exploit technological advances which offer opportunities for improving the educational process. One area of technology which offers opportunities for enhancing the manner in which research is conducted and ultimately affects scientific and engineering education is computer networks. As computer hardware has become less expensive, more numerous and more capable, individuals and organizations have developed a keen interest in connecting them together in order to form networks. This in turn has had an impact on the manner in which laboratory research is conducted. This paper addresses a relatively new approach to scientific research, telescience, which is the conduct of scientific operations in locations remote from the site of central experimental activity. A testbed based on the concepts of telescience is being developed to ultimately enable scientific researchers on earth to conduct experiments onboard the Space Station. This system along with background materials are discussed in this paper.

Tody, D., "The IRAF Data Reduction and Analysis System," National Optical Astronomical Observatories, Tucson, AZ, 1986.

Walker, M. T., S-Y Sheu, R. Volz, and L. Conway, "A Low Cost Portable Tele-Autonomous Maintenance Station," *SOAR 88: A Workshop on Space Applications of Artificial Intelligence, Human Factors and Robotics*, Wright State University, Dayton, OH, July 20-23, 1988.

Walker, W. T., *Video Data Compression for Telescience*, Technical Report TSL-017/88, Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, September 1988.

White, O.R. and G.J. Rottman, *SME as a Testbed for Telescience - A Case Study*, LASP Report 89-2, University of Colorado, Boulder, CO, February 1989.

Wiskerchen, Michael J. and Barry M. Leiner, "Telescience Testbed Pilot Project: Evaluation Environment for Space Station Operations," AIAA'88, AIAA Flight Simulation Tech., Atlanta, GA, September 1988

This paper describes the structure and methodology of the rapid prototyping efforts and reports the results for the first eleven months of the 15 university telescience testbed program. In addition, the multi-media networking capabilities between the NASA Centers involved in space station design and operations and the universities are discussed in terms of overall requirements for telecommunications between space station testbed/simulation facilities and the telescience testbed effort.

Young, Larry A. and Barry M. Leiner, "Telescience," AIAA/NASA First International Symposium on Space Automation and Robotics, November 1988, (also available as RIACS TR 88.28, 9 pp., (MIT) Young, (RIACS) Leiner, October 1988).

Telescience is the approach and collection of tools that enable productive scientific activity to be carried out using remote resources. By using interactive high-performance telecommunication links between space-based laboratories and facilities, on-orbit crew, and geographically dispersed ground-based investigator groups, facilities such as Space Station become an accessible and integral part of the research environment. In this paper, we describe an innovative program of rapid prototyping testbeds aimed at evaluating and validating telescience modes of operation and the technologies to support them. Particular attention is given to three testbeds evaluating remote instrumentation monitoring and control, expert systems in support of the interaction between the principal investigator and the astronaut, and telerobotics in support of fluid handling. In all of the testbeds, the application of these new technologies have been shown to improve scientific productivity.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Index

A

ACCESS III-15
 Access III-67, III-76, III-78
 access III-27
 Ada III-3, III-15, III-17, III-33, III-42
 Ames III-18, III-23, III-33, III-34, III-78
 Ames Research Center, see ARC
 ARC III-118
 Arizona III-1, III-13, III-18, III-22, III-27, III-34, III-56, III-70, III-99, III-103, III-116
 ARPAnet III-53, III-62
 Astrometric Telescope III-18
 Astrometric Telescope Facility III-13
 Astronomy III-30
 ATF III-13
 ATHENA III-43, III-75
 AVHRR III-78, III-91, III-92, III-94

B

bandwidth III-100
 BARRnet III-62
 BDCF III-44
 BEARS III-86
 Berkeley III-8, III-10, III-59, III-62, III-84, III-86, III-103, III-117
 BITnet III-53
 Bitnet III-62
 BROWSE III-89
 Browse III-84, III-87
 browse III-91

C

C III-27, III-75, III-81
 Cal Tech III-119
 Calen III-102
 California Institute of Technology, see Cal Tech
 Campaign III-65
 catalog III-76
 CCD III-58
 CCSDS III-14, III-19, III-40
 CLIPS III-6, III-97
 coaching III-5, III-44, III-95, III-96
 Colorado III-3, III-14, III-18, III-30, III-33, III-36, III-39, III-53, III-60, III-65, III-67, III-93, III-111, III-117
 Command Interlock III-36
 Communications III-95

Compaq III-97
 compression III-18, III-63
 Cornell III-1, III-68, III-103, III-115
 CSTOL III-34

D

database III-67, III-76, III-82, III-93
 databases III-89
 DECnet III-14, III-19
 delay III-16
 delays III-23
 DESQView III-97
 DIF III-41
 Documents III-99
 DOS III-97
 DVI III-102

E

e-mail III-98
 earth science III-76, III-92
 earth system science III-65
 Electronic mail III-98
 electronic mail III-60, III-67, III-110
 Emulation III-10
 Eunice III-61
 EUVE III-60, III-84
 expert system III-5, III-7, III-95

F

field campaigns III-89
 file transfer III-60, III-88
 FITS III-2, III-70
 Fluid Handling III-22
 formats III-1
 Frame Maker III-102
 frame rate III-49
 FRG III-49
 FTP III-79, III-82, III-94

G

GKS III-88
 Glovebox III-24
 GNU Emacs III-102
 GO III-84
 Goddard III-80, III-103, III-111
 Goddard Space Flight Center III-78, III-80, III-91
 GOES III-89, III-92
 grey scale III-49
 GSFC III-99
 Guest Observer III-84

H

HyperCard III-76

I

IBM AT III-97
 IBM PC III-60, III-93, III-97
 IBM PC XT III-65
 IBM PC/AT III-27, III-87
 IBM PS/2 50 III-91
 Image III-90
 image III-75
 Images III-27
 images III-2, III-79
 infrared array III-69
 interface III-101
 Internet III-1, III-56, III-62, III-78, III-88, III-99
 IR array III-57, III-58
 IRAF III-1, III-28, III-29, III-69, III-85
 IRIS III-112

J

Johnson Space Center, see JSC
 JSC III-103, III-112, III-119

K

Kansas III-78
 KAO III-71
 Kennedy Space Center, see KSC
 KERMIT III-95
 Kermit III-28, III-70, III-88, III-102
 Kiwi III-60
 KSC III-43, III-119
 KSC Baseline Data Collection Facility III-44
 KSC BDCF III-46

L

LAN III-62
 Landsat 5 III-89
 LaTeX III-102
 LeRC III-51, III-119
 Lewis Research Center III-51
 Lewis Research Center, see LeRC
 Life Sciences III-22, III-43
 Lowell Observatory III-72

M

Macintosh III-82, III-102
 Magic/L III-11

Maryland III-103, III-117
 Massachusetts Institute of Technology, see MIT
 Mauna Kea III-75
 McIDAS III-78, III-89, III-92, III-93
 meteorological data III-92
 Miami III-91
 Michigan III-5, III-7, III-95, III-96, III-118
 Microgravity Materials Science III-51
 Microgravity Sciences III-22
 microVAX III-14, III-18, III-56, III-60
 MIT III-8, III-11, III-43, III-71, III-115
 modems III-77
 monitor III-35
 monitoring III-111
 Mt. Hopkins III-57, III-59

N

NASAselect III-43
 Network III-99
 network III-68
 networking III-110
 Networks III-98
 networks III-17, III-53, III-61, III-77, III-88
 NFS III-1, III-29
 NSFnet III-17, III-62, III-93
 NSI III-10, III-61, III-100
 NSN III-17, III-21, III-57, III-99

O

OASIS III-3, III-14, III-18, III-23, III-29, III-30,
 III-33, III-53, III-111
 ocular torsion III-43
 Off-line Guidance III-5
 Olivetti III-103
 OMS III-16, III-36
 On-line Guidance III-7
 optical disc III-75
 Optical Disk III-82
 optical disks III-67, III-77
 optokinetic nystagmus III-43
 Oracle III-77

P

packet telemetry III-40
 Parallax III-50
 Participants III-115
 Payload Systems Inc. III-43
 Payload Systems, Inc. III-115
 PC-AT III-79
 Pittsburgh III-18
 Portability III-3

portability III-69, III-111
 productivity III-74
 programmatic III-110
 Programmatics III-98
 PSI III-43
 Purdue III-34, III-76, III-78, III-82, III-90, III-92, III-93, III-103, III-115

R

Rand MH III-102
 rate III-18
 rates III-3, III-90
 Remote Fluid Handling III-13
 Rensselaer Polytechnic Institute, see RPI
 Research Institute for Advanced Computer Science,
 see RIACS
 Resolution III-90
 resolution III-49
 resource envelopes III-37
 RFH III-13
 Rhode Island III-90, III-118
 RIACS III-53, III-101, III-109, III-118
 robot III-22
 Rochester III-68, III-119
 RPI III-51, III-116

S

safety III-20, III-30
 Santa Barbara III-78, III-84, III-87, III-89, III-117
 SAO III-56, III-99, III-103, III-116
 SCS III-8
 security III-61
 SF DU III-42
 SIMULATION III-16
 Simulation III-80
 simulation III-19
 simulator III-60
 SIRTF III-1, III-56, III-68, III-118
 SME III-33, III-36, III-40, III-65, III-67, III-89
 Smithsonian Astrophysical Observatory III-56, III-98, III-99
 Smithsonian Astrophysical Observatory, see SAO
 Software III-1
 software III-101, III-111
 Software Control System III-8
 Software Interlock III-36
 Solar Mesosphere Explorer III-40
 Sommers-Bausch telescope III-32
 Space Telescope Science Institute III-103
 Spacelab III-80
 SPAN III-41, III-62, III-80, III-88, III-91
 SSIS III-44

standardization III-109
 standards III-28, III-68
 Stanford III-34, III-60, III-80, III-103, III-111,
 III-116
 Steward Observatory III-27, III-57
 Sun III-1, III-3, III-11, III-27, III-35, III-53, III-
 69, III-83, III-86, III-101, III-103, III-111

T

TAE III-111
 TAE+ III-29
 TCP/IP III-1, III-28, III-61, III-62, III-80, III-
 81, III-93, III-95
 TDRS III-41
 Teledesign III-1
 Telemetry III-42, III-59
 telemetry III-87
 Teleoperation III-13, III-18
 telescope III-27
 TeleWEn III-101
 TeX III-102
 Thaw telescope III-18
 Transaction Management III-36

U

UCB, see Berkeley
 UCSB III-84, III-89, III-93
 UCSB, see Santa Barbara
 Ultrix III-111
 Universities Space Research Association, see USRA
 UNIX III-1, III-3, III-28, III-35, III-50, III-75,
 III-86, III-98, III-103
 Unix III-111
 Usenet III-62
 User Interface III-29
 user interface III-32
 USRA III-118
 Utah graphics tool kit III-102

V

VAX III-80, III-111
 VERDIX III-4
 Video III-43
 video III-17, III-23, III-60
 VM-CMS III-76
 VMS III-61, III-86, III-111
 voice III-44

W

Wisconsin III-78, III-90, III-92, III-118

Workstation III-101
Workstations III-96
workstations III-1, III-111

X

X III-50, III-58
X-ray Timing Explorer III-8
XENIX III-98
XMODEM III-95
XTE III-8

Y

YMODEM III-95

Z

ZMODEM III-95



Mail Stop 230-5
NASA Ames Research Center
Moffett Field, CA 94035
(415) 694-6363

The Research Institute for Advanced Computer Science
is an Institute of the
Universities Space Research Association
The American City Building
Suite 212
Columbia, MD 21044
(301) 730-2656